```
better-sends-rule  THM

⊢  ∀i:Id, k:Knd, l:IdLnk, ds:x:Id fp-> Type, da:a:Knd fp-> Type,
|   f:(tg:Id
|      ×State(ds)→ma-valtype(da; k)→
|      (da(rcv(l; tg))?Void List)) List.
|  source(l) = i
|  ⇒
|  d-single-sends(i; ds; da; k; l; f)
|  realizes es.(∀x:Id. vartype(i;x) ⊆r ds(x)?Top)
|              & (∀e:E.
|                 loc(e) = i
|                 ⇒
|                 (valtype(e) ⊆r ma-valtype(da; kind(e))))
|              & (∀e:E.
|                 isrcv(e)
|                 ⇒
|                 lnk(e) = l
|                 ⇒
|                 (valtype(e) ⊆r ma-valtype(da; kind(e))))
|              & (∀e:E.
|                 loc(e) = i
|                 ⇒
|                 kind(e) = k
|                 ⇒
|                 (∃L:{e':E| isrcv(e') & lnk(e') = l } List.
|                  (∀e':E.
|                   (e' ∈ L)
|                   ⟺
|                   isrcv(e') & lnk(e') = l & sender(e') = e)
|                  & (∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2))
|                  & map(λe'.<tag(e'),val(e')>;L)
|                    =
|                    tagged-list-messages(λz.(z when e);val(e);f)))
| by Inst Thm: sends_DASH_rule [] THEN RepeatFor 13 ParallelLast
|    THEN
|    ExRepD
|    THEN
|    BetterSplitAndConcl
|    THEN
|    Try Trivial
|    THEN
|    ExRepD
|    THEN
|    AllHyps (λh.InstHyp [e] h THENA Complete Auto)
|
1. i : Id
2. k : Knd
3. l : IdLnk
4. ds : x:Id fp-> Type
5. da : a:Knd fp-> Type
6. f :
|  (tg:Id
|   ×State(ds)→ma-valtype(da; k)→(da(rcv(l; tg))?Void List)) List
7. source(l) = i
```

```
8. D' : Dsys
9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
10. w : World
11. p : FairFifo
12. PossibleWorld(D';w)
13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
15. ∀e:E.
|   isrcv(e) ⇒ lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg((λl,tg. da(rcv(l; tg))?Top))
17. ∀e:E.
|   loc(e) = i
|   ⇒
|   kind(e) = k
|   ⇒
|   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
20. ∀e:E.
|   isrcv(e) ⇒ lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
21. e : E
22. loc(e) = i
23. kind(e) = k
24. valtype(e) ⊆r ma-valtype(da; kind(e))
25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
26. valtype(e) ⊆r ma-valtype(da; kind(e))
⊢  ∃L:{e':E| isrcv(e') & lnk(e') = l } List.
|  (∀e':E. (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e)
|  & (∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2))
|  & map(λe'.<tag(e'),val(e')>;L)
|    =
|    tagged-list-messages(λz.(z when e);val(e);f)
| by InstESAxiom ES(w) [e;l] 16 THEN RenameVar ‘L’ -1
|    THEN
|    InstConcl [map(λn.1of(L(n));upto(||sends(l;e)||))]
|
|\
| .....wf .....
| 1. i : Id
| 2. k : Knd
| 3. l : IdLnk
| 4. ds : x:Id fp-> Type
| 5. da : a:Knd fp-> Type
| 6. f :
| |  (tg:Id
| |   ×State(ds)→ma-valtype(da; k)→(da(rcv(l; tg))?Void List)) List
| 7. source(l) = i

| 8. D' : Dsys
| 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| 10. w : World
| 11. p : FairFifo
| 12. PossibleWorld(D';w)
| 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| 15. ∀e:E.
```

```
| |   isrcv(e)
| |   ⇒
| |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| |   ((λl,tg. da(rcv(l; tg))?Top))
| 17. ∀e:E.
| |   loc(e) = i
| |   ⇒
| |   kind(e) = k
| |   ⇒
| |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| 20. ∀e:E.
| |   isrcv(e)
| |   ⇒
| |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| 21. e : E
| 22. loc(e) = i
| 23. kind(e) = k
| 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| 27. L :
| |   ∀n:ℕ||sends(l;e)||.
| |   ∃e':E. isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| ⊢  map(λn.1of(L(n));upto(||sends(l;e)||))
| |  ∈ {e':E| isrcv(e') & lnk(e') = l } List  by Auto
| |
| .....subterm .....
| 1. i : Id
| 2. k : Knd
| 3. l : IdLnk
| 4. ds : x:Id fp-> Type
| 5. da : a:Knd fp-> Type
| 6. f :
|    (tg:Id
|     ×State(ds)→ma-valtype(da; k)→(da(rcv(l; tg))?Void List)) List
| 7. source(l) = i
| 8. D' : Dsys
| 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| 10. w : World
| 11. p : FairFifo
| 12. PossibleWorld(D';w)
| 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| 15. ∀e:E.
|     isrcv(e)
|     ⇒
|     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
|     ((λl,tg. da(rcv(l; tg))?Top))
| 17. ∀e:E.
|     loc(e) = i
|     ⇒
```

```
|     kind(e) = k
|     ⇒
|     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| 20. ∀e:E.
|     isrcv(e)
|     ⇒
|     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| 21. e : E
| 22. loc(e) = i
| 23. kind(e) = k
| 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| 27. L :
|     ∀n:ℕ||sends(l;e)||.
|     ∃e':E. isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| 28. n : ℕ||sends(l;e)||
| ⊢  L(n)
|    ∈ a:{e':E| isrcv(e') & lnk(e') = l }
|      ×isrcv(a) & lnk(a) = l & sender(a) = e & index(a) = n
|   by DoSubsume THEN Analyze 0 THEN Analyze -1 THEN ExRepD
|      THEN
|      Analyze
|
|\
| 1. i : Id
| 2. k : Knd
| 3. l : IdLnk
| 4. ds : x:Id fp-> Type
| 5. da : a:Knd fp-> Type
| 6. f :
| |  (tg:Id
| |   ×State(ds)→ma-valtype(da; k)→(da(rcv(l; tg))?Void List)) List
| 7. source(l) = i
| 8. D' : Dsys
| 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| 10. w : World
| 11. p : FairFifo
| 12. PossibleWorld(D';w)
| 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| 15. ∀e:E.
| |   isrcv(e)
| |   ⇒
| |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| |   ((λl,tg. da(rcv(l; tg))?Top))
| 17. ∀e:E.
| |   loc(e) = i
| |   ⇒
| |   kind(e) = k
| |   ⇒
| |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
```

```
| 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| 20. ∀e:E.
| |   isrcv(e)
| |   ⇒
| |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| 21. e : E
| 22. loc(e) = i
| 23. kind(e) = k
| 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| 27. L :
| |   ∀n:ℕ||sends(l;e)||.
| |   ∃e':E. isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| ⊢  (∀e':E.
| |   (e' ∈ map(λn.1of(L(n));upto(||sends(l;e)||)))
| |   ⟺
| |   isrcv(e') & lnk(e') = l & sender(e') = e)
| |  & (∀e1,e2:E.
| |     e1 before e2 ∈ map(λn.1of(L(n));upto(||sends(l;e)||))
| |     ⇒
| |     (e1 <loc e2))
| |  & map(λe'.<tag(e'),val(e')>;map(λn.
| |    1of(L(n));upto(||sends(l;e)||)))
| |    =
| |    tagged-list-messages(λz.(z when e);val(e);f)
| | by GenConcl ((λn.1of(L(n))) = LL)
| |
| |\
| | .....wf .....
| | 1. i : Id
| | 2. k : Knd
| | 3. l : IdLnk
| | 4. ds : x:Id fp-> Type
| | 5. da : a:Knd fp-> Type
| | 6. f :
| |    (tg:Id
| |     ×State(ds)→ma-valtype(da; k)→
| |     (da(rcv(l; tg))?Void List)) List
| | 7. source(l) = i
| | 8. D' : Dsys
| | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| | 10. w : World
| | 11. p : FairFifo
| | 12. PossibleWorld(D';w)
| | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| | 15. ∀e:E.
| |     isrcv(e)
| |     ⇒
| |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| |     ((λl,tg. da(rcv(l; tg))?Top))
| | 17. ∀e:E.
```

```
| |     loc(e) = i
| |     ⇒
| |     kind(e) = k
| |     ⇒
| |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| | 20. ∀e:E.
| |     isrcv(e)
| |     ⇒
| |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| | 21. e : E
| | 22. loc(e) = i
| | 23. kind(e) = k
| | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| | 27. L :
| |     ∀n:ℕ||sends(l;e)||.
| |     ∃e':E.
| |     isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| | ⊢  (λn.1of(L(n)))
| |    ∈ n:ℕ||sends(l;e)||→{e':E
| |                          | isrcv(e')
| |                            & lnk(e') = l
| |                              & sender(e') = e
| |                              & index(e') = n }

| |   by Auto THEN DoSubsume THEN Analyze 0 THEN Analyze -1
| |
| |\
| | .....wf .....
| | 1. i : Id
| | 2. k : Knd
| | 3. l : IdLnk
| | 4. ds : x:Id fp-> Type
| | 5. da : a:Knd fp-> Type
| | 6. f :
| |    (tg:Id
| |     ×State(ds)→ma-valtype(da; k)→
| |     (da(rcv(l; tg))?Void List)) List
| | 7. source(l) = i
| | 8. D' : Dsys
| | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| | 10. w : World
| | 11. p : FairFifo
| | 12. PossibleWorld(D';w)
| | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| | 15. ∀e:E.
| |     isrcv(e)
| |     ⇒
| |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| |     ((λl,tg. da(rcv(l; tg))?Top))
| | 17. ∀e:E.
```

```
| |     loc(e) = i
| |     ⇒
| |     kind(e) = k
| |     ⇒
| |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| | 20. ∀e:E.
| |     isrcv(e)
| |     ⇒
| |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| | 21. e : E
| | 22. loc(e) = i
| | 23. kind(e) = k
| | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| | 27. ∀n:ℕ||sends(l;e)||.
| |     ∃e':E.
| |     isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| | ⊢  n:ℕ||sends(l;e)||→{e':E
| |                       | isrcv(e')
| |                         & lnk(e') = l
| |                           & sender(e') = e
| |                           & index(e') = n }
| |    ∈ Type  by Auto
| |
| .\
| . 1. i : Id
| . 2. k : Knd
| . 3. l : IdLnk
| . 4. ds : x:Id fp-> Type
| . 5. da : a:Knd fp-> Type
| . 6. f :
| . |  (tg:Id
| . |   ×State(ds)→ma-valtype(da; k)→
| . |   (da(rcv(l; tg))?Void List)) List
| . 7. source(l) = i
| . 8. D' : Dsys
| . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . 10. w : World
| . 11. p : FairFifo
| . 12. PossibleWorld(D';w)
| . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . 15. ∀e:E.
| . |   isrcv(e)
| . |   ⇒
| . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . |   ((λl,tg. da(rcv(l; tg))?Top))
| . 17. ∀e:E.
| . |   loc(e) = i
| . |   ⇒
| . |   kind(e) = k
```

```
| . |   ⇒
| . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . 20. ∀e:E.
| . |   isrcv(e)
| . |   ⇒
| . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . 21. e : E
| . 22. loc(e) = i
| . 23. kind(e) = k
| . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . 27. L :
| . |   ∀n:ℕ||sends(l;e)||.
| . |   ∃e':E.
| . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . 28. LL :
| . |   n:ℕ||sends(l;e)||→{e':E
| . |                      | isrcv(e')
| . |                        & lnk(e') = l
| . |                          & sender(e') = e
| . |                          & index(e') = n }
| . 29. (λn.1of(L(n))) = LL
| . ⊢  (∀e':E.
| . |   (e' ∈ map(LL;upto(||sends(l;e)||)))
| . |   ⟺
| . |   isrcv(e') & lnk(e') = l & sender(e') = e)
| . |  & (∀e1,e2:E.
| . |     e1 before e2 ∈ map(LL;upto(||sends(l;e)||)) ⇒ (e1 <loc e2))
| . |  & map(λe'.<tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))
| . |    =
| . |    tagged-list-messages(λz.(z when e);val(e);f)
| . | by SplitAndConcl
| . |
| . |\
| . | 1. i : Id
| . | 2. k : Knd
| . | 3. l : IdLnk
| . | 4. ds : x:Id fp-> Type
| . | 5. da : a:Knd fp-> Type
| . | 6. f :
| . | |  (tg:Id
| . | |   ×State(ds)→ma-valtype(da; k)→
| . | |   (da(rcv(l; tg))?Void List)) List
| . | 7. source(l) = i
| . | 8. D' : Dsys
| . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | 10. w : World
| . | 11. p : FairFifo
| . | 12. PossibleWorld(D';w)
| . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 15. ∀e:E.
```

```
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . | 17. ∀e:E.
| . | |   loc(e) = i
| . | |   ⇒
| . | |   kind(e) = k
| . | |   ⇒
| . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 20. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 21. e : E
| . | 22. loc(e) = i
| . | 23. kind(e) = k
| . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 27. L :
| . | |   ∀n:ℕ||sends(l;e)||.
| . | |   ∃e':E.
| . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | 28. LL :
| . | |   n:ℕ||sends(l;e)||→{e':E
| . | |                      | isrcv(e')
| . | |                        & lnk(e') = l
| . | |                          & sender(e') = e
| . | |                          & index(e') = n }
| . | 29. (λn.1of(L(n))) = LL
| . | ⊢  ∀e':E.
| . | |  (e' ∈ map(LL;upto(||sends(l;e)||)))
| . | |  ⟺
| . | |  isrcv(e') & lnk(e') = l & sender(e') = e
| . | | by RepeatFor 3 (Analyze 0)
| . | |
| . | |\
| . | | 1. i : Id
| . | | 2. k : Knd
| . | | 3. l : IdLnk
| . | | 4. ds : x:Id fp-> Type
| . | | 5. da : a:Knd fp-> Type
| . | | 6. f :
| . | | |  (tg:Id
| . | | |   ×State(ds)→ma-valtype(da; k)→
| . | | |   (da(rcv(l; tg))?Void List)) List
| . | | 7. source(l) = i
| . | | 8. D' : Dsys
| . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | | 10. w : World
| . | | 11. p : FairFifo
```

```
| . | | 12. PossibleWorld(D';w)
| . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | | 14. ∀e:E.
| . | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | | 15. ∀e:E.
| . | | |   isrcv(e)
| . | | |   ⇒
| . | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . | | 17. ∀e:E.
| . | | |   loc(e) = i
| . | | |   ⇒
| . | | |   kind(e) = k
| . | | |   ⇒
| . | | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | | 19. ∀e:E.
| . | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | | 20. ∀e:E.
| . | | |   isrcv(e)
| . | | |   ⇒
| . | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | | 21. e : E
| . | | 22. loc(e) = i
| . | | 23. kind(e) = k
| . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | | 27. L :
| . | | |   ∀n:ℕ||sends(l;e)||.
| . | | |   ∃e':E.
| . | | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | | 28. LL :
| . | | |   n:ℕ||sends(l;e)||→{e':E
| . | | |                      | isrcv(e')
| . | | |                        & lnk(e') = l
| . | | |                          & sender(e') = e
| . | | |                          & index(e') = n }
| . | | 29. (λn.1of(L(n))) = LL
| . | | 30. e' : E
| . | | 31. (e' ∈ map(LL;upto(||sends(l;e)||)))
| . | | ⊢  isrcv(e') & lnk(e') = l & sender(e') = e
| . | | | by RWO
| . | | |    Thm* ∀a:T List, x:T', f:(T→T').
| . | | |         (x ∈ map(f;a)) ⟺ (∃y:T. (y ∈ a) & x = f(y))
| . | | |    {member_map}
| . | | |    -1
| . | | |    THEN
| . | | |    ExRepD
| . | | |    THEN
| . | | |    HypSubst -1 0
| . | | |
| . | | 1. i : Id
| . | | 2. k : Knd
```

```
| . | | 3. l : IdLnk
| . | | 4. ds : x:Id fp-> Type
| . | | 5. da : a:Knd fp-> Type
| . | | 6. f :
| . | |    (tg:Id
| . | |     ×State(ds)→ma-valtype(da; k)→
| . | |     (da(rcv(l; tg))?Void List)) List
| . | | 7. source(l) = i
| . | | 8. D' : Dsys
| . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | | 10. w : World
| . | | 11. p : FairFifo
| . | | 12. PossibleWorld(D';w)
| . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | | 14. ∀e:E.
| . | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | | 15. ∀e:E.
| . | |     isrcv(e)
| . | |     ⇒
| . | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | |     ((λl,tg. da(rcv(l; tg))?Top))
| . | | 17. ∀e:E.
| . | |     loc(e) = i
| . | |     ⇒
| . | |     kind(e) = k
| . | |     ⇒
| . | |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | | 19. ∀e:E.
| . | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | | 20. ∀e:E.
| . | |     isrcv(e)
| . | |     ⇒
| . | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | | 21. e : E
| . | | 22. loc(e) = i
| . | | 23. kind(e) = k
| . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | | 27. L :
| . | |     ∀n:ℕ||sends(l;e)||.
| . | |     ∃e':E.
| . | |     isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | | 28. LL :
| . | |     n:ℕ||sends(l;e)||→{e':E
| . | |                        | isrcv(e')
| . | |                          & lnk(e') = l
| . | |                            & sender(e') = e
| . | |                            & index(e') = n }
| . | | 29. (λn.1of(L(n))) = LL

| . | | 30. e' : E
| . | | 31. y : ℕ||sends(l;e)||
| . | | 32. (y ∈ upto(||sends(l;e)||))
```

```
| . | | 33. e' = LL(y)
| . | | ⊢  isrcv(LL(y)) & lnk(LL(y)) = l & sender(LL(y)) = e
| . | |   by GenConclAtAddr [1;1;2] THEN Analyze -2 THEN Unhide
| . | |      THEN
| . | |      ExRepD
| . | |
| . | .\
| . | . 1. i : Id
| . | . 2. k : Knd
| . | . 3. l : IdLnk
| . | . 4. ds : x:Id fp-> Type
| . | . 5. da : a:Knd fp-> Type
| . | . 6. f :
| . | . |  (tg:Id
| . | . |   ×State(ds)→ma-valtype(da; k)→
| . | . |   (da(rcv(l; tg))?Void List)) List
| . | . 7. source(l) = i
| . | . 8. D' : Dsys
| . | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | . 10. w : World
| . | . 11. p : FairFifo
| . | . 12. PossibleWorld(D';w)
| . | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . 14. ∀e:E.
| . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 15. ∀e:E.
| . | . |   isrcv(e)
| . | . |   ⇒
| . | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . | . 17. ∀e:E.
| . | . |   loc(e) = i
| . | . |   ⇒
| . | . |   kind(e) = k
| . | . |   ⇒
| . | . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . 19. ∀e:E.
| . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 20. ∀e:E.
| . | . |   isrcv(e)
| . | . |   ⇒
| . | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 21. e : E
| . | . 22. loc(e) = i
| . | . 23. kind(e) = k
| . | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . 27. L :
| . | . |   ∀n:ℕ||sends(l;e)||.
| . | . |   ∃e':E.
| . | . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | . 28. LL :
```

```
| . | . |   n:ℕ||sends(l;e)||→{e':E
| . | . |                      | isrcv(e')
| . | . |                        & lnk(e') = l
| . | . |                          & sender(e') = e
| . | . |                          & index(e') = n }
| . | . 29. (λn.1of(L(n))) = LL
| . | . 30. e' : E
| . | . 31. isrcv(e') & lnk(e') = l & sender(e') = e
| . | . ⊢  (e' ∈ map(LL;upto(||sends(l;e)||)))
| . | . | by ExRepD
| . | . |    THEN
| . | . |    RWO
| . | . |    Thm* ∀a:T List, x:T', f:(T→T').
| . | . |         (x ∈ map(f;a)) ⟺ (∃y:T. (y ∈ a) & x = f(y))
| . | . |    {member_map}
| . | . |    0
| . | . |
| . | . 1. i : Id
| . | . 2. k : Knd
| . | . 3. l : IdLnk
| . | . 4. ds : x:Id fp-> Type
| . | . 5. da : a:Knd fp-> Type
| . | . 6. f :
| . | . |  (tg:Id
| . | . |   ×State(ds)→ma-valtype(da; k)→
| . | . |   (da(rcv(l; tg))?Void List)) List
| . | . 7. source(l) = i
| . | . 8. D' : Dsys
| . | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | . 10. w : World
| . | . 11. p : FairFifo
| . | . 12. PossibleWorld(D';w)
| . | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . 14. ∀e:E.
| . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 15. ∀e:E.
| . | . |   isrcv(e)
| . | . |   ⇒
| . | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . | . 17. ∀e:E.
| . | . |   loc(e) = i
| . | . |   ⇒
| . | . |   kind(e) = k
| . | . |   ⇒
| . | . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . 19. ∀e:E.
| . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 20. ∀e:E.
| . | . |   isrcv(e)
| . | . |   ⇒
| . | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 21. e : E
```

```
| . | . 22. loc(e) = i
| . | . 23. kind(e) = k
| . | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . 27. L :
| . | . |   ∀n:ℕ||sends(l;e)||.
| . | . |   ∃e':E.
| . | . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | . 28. LL :
| . | . |   n:ℕ||sends(l;e)||→{e':E
| . | . |                       | isrcv(e')
| . | . |                         & lnk(e') = l
| . | . |                           & sender(e') = e
| . | . |                           & index(e') = n }
| . | . 29. (λn.1of(L(n))) = LL
| . | . 30. e' : E
| . | . 31. isrcv(e')
| . | . 32. lnk(e') = l
| . | . 33. sender(e') = e
| . | . ⊢  ∃y:ℕ||sends(l;e)||. (y ∈ upto(||sends(l;e)||)) & e' = LL(y)
| . | . | by AssertBY (index(e') ∈ ℕ||sends(l;e)||)
| . | . |    (DoSubsume THEN HypSubst' -2 0 THEN HypSubst -1 0)
| . | . |    THEN
| . | . |    InstConcl [index(e')]
| . | . |    THEN
| . | . |    Try
| . | . |    (BackThru
| . | . |     Thm* ∀n,i:ℕ. i<n ⇒ (i ∈ upto(n))    {member_upto2})
| . | . |
| . | . 1. i : Id
| . | . 2. k : Knd
| . | . 3. l : IdLnk
| . | . 4. ds : x:Id fp-> Type
| . | . 5. da : a:Knd fp-> Type
| . | . 6. f :
| . | . |  (tg:Id
| . | . |   ×State(ds)→ma-valtype(da; k)→
| . | . |   (da(rcv(l; tg))?Void List)) List
| . | . 7. source(l) = i
| . | . 8. D' : Dsys
| . | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | . 10. w : World
| . | . 11. p : FairFifo
| . | . 12. PossibleWorld(D';w)
| . | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . 14. ∀e:E.
| . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 15. ∀e:E.
| . | . |   isrcv(e)
| . | . |   ⇒
| . | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . | . 17. ∀e:E.
```

```
| . | . |   loc(e) = i
| . | . |   ⇒
| . | . |   kind(e) = k
| . | . |   ⇒
| . | . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . 19. ∀e:E.
| . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 20. ∀e:E.
| . | . |   isrcv(e)
| . | . |   ⇒
| . | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 21. e : E
| . | . 22. loc(e) = i
| . | . 23. kind(e) = k
| . | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . 27. L :
| . | . |   ∀n:ℕ||sends(l;e)||.
| . | . |   ∃e':E.
| . | . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | . 28. LL :
| . | . |   n:ℕ||sends(l;e)||→{e':E
| . | . |                       | isrcv(e')
| . | . |                         & lnk(e') = l
| . | . |                           & sender(e') = e
| . | . |                           & index(e') = n }
| . | . 29. (λn.1of(L(n))) = LL
| . | . 30. e' : E
| . | . 31. isrcv(e')
| . | . 32. lnk(e') = l
| . | . 33. sender(e') = e
| . | . 34. index(e') ∈ ℕ||sends(l;e)||
| . | . ⊢  e' = LL(index(e'))
| . | . | by GenConclAtAddr [3] THEN Analyze -2 THEN Unhide
| . | . |    THEN
| . | . |    ExRepD
| . | . |
| . | . 1. i : Id
| . | . 2. k : Knd
| . | . 3. l : IdLnk
| . | . 4. ds : x:Id fp-> Type
| . | . 5. da : a:Knd fp-> Type
| . | . 6. f :
| . | . |  (tg:Id
| . | . |   ×State(ds)→ma-valtype(da; k)→
| . | . |   (da(rcv(l; tg))?Void List)) List
| . | . 7. source(l) = i
| . | . 8. D' : Dsys
| . | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | . 10. w : World
| . | . 11. p : FairFifo
| . | . 12. PossibleWorld(D';w)
| . | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
```

```
| . | . 14. ∀e:E.
| . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 15. ∀e:E.
| . | . |   isrcv(e)
| . | . |   ⇒
| . | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . | . 17. ∀e:E.
| . | . |   loc(e) = i
| . | . |   ⇒
| . | . |   kind(e) = k
| . | . |   ⇒
| . | . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . 19. ∀e:E.
| . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 20. ∀e:E.
| . | . |   isrcv(e)
| . | . |   ⇒
| . | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . 21. e : E
| . | . 22. loc(e) = i
| . | . 23. kind(e) = k
| . | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . 27. L :
| . | . |   ∀n:ℕ||sends(l;e)||.
| . | . |   ∃e':E.
| . | . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | . 28. LL :
| . | . |   n:ℕ||sends(l;e)||⟶{e':E
| . | . |                      | isrcv(e')
| . | . |                        & lnk(e') = l
| . | . |                          & sender(e') = e
| . | . |                          & index(e') = n }
| . | . 29. (λn.1of(L(n))) = LL
| . | . 30. e' : E
| . | . 31. isrcv(e')
| . | . 32. lnk(e') = l
| . | . 33. sender(e') = e
| . | . 34. index(e') ∈ ℕ||sends(l;e)||
| . | . 35. v : E
| . | . 36. isrcv(v)
| . | . 37. lnk(v) = l
| . | . 38. sender(v) = e
| . | . 39. index(v) = index(e')
| . | . 40. LL(index(e')) = v
| . | . ⊢  e' = v
| . | . | by InstESAxiom ES(w) [e';v] 3 THEN Analyze -1 THEN Thin -1
| . | . |    THEN
| . | . |    Analyze -1
| . | . |
| . | . |\
```

```
| . | . | .....antecedent .....
| . | . | 1. i : Id
| . | . | 2. k : Knd
| . | . | 3. l : IdLnk
| . | . | 4. ds : x:Id fp-> Type
| . | . | 5. da : a:Knd fp-> Type
| . | . | 6. f :
| . | . |    (tg:Id
| . | . |     ×State(ds)→ma-valtype(da; k)→
| . | . |     (da(rcv(l; tg))?Void List)) List
| . | . | 7. source(l) = i
| . | . | 8. D' : Dsys
| . | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | . | 10. w : World
| . | . | 11. p : FairFifo
| . | . | 12. PossibleWorld(D';w)
| . | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . | 14. ∀e:E.
| . | . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . | 15. ∀e:E.
| . | . |     isrcv(e)
| . | . |     ⇒
| . | . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | . |     ((λl,tg. da(rcv(l; tg))?Top))
| . | . | 17. ∀e:E.
| . | . |     loc(e) = i
| . | . |     ⇒
| . | . |     kind(e) = k
| . | . |     ⇒
| . | . |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . | 19. ∀e:E.
| . | . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . | 20. ∀e:E.
| . | . |     isrcv(e)
| . | . |     ⇒
| . | . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . | 21. e : E
| . | . | 22. loc(e) = i
| . | . | 23. kind(e) = k
| . | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . | 27. L :
| . | . |     ∀n:ℕ||sends(l;e)||.
| . | . |     ∃e':E.
| . | . |     isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | . | 28. LL :
| . | . |     n:ℕ||sends(l;e)||→{e':E
| . | . |                       | isrcv(e')
| . | . |                         & lnk(e') = l
| . | . |                           & sender(e') = e
| . | . |                           & index(e') = n }
| . | . | 29. (λn.1of(L(n))) = LL
```

```
| . | . | 30. e' : E
| . | . | 31. isrcv(e')
| . | . | 32. lnk(e') = l
| . | . | 33. sender(e') = e
| . | . | 34. index(e') ∈ ℕ||sends(l;e)||
| . | . | 35. v : E
| . | . | 36. isrcv(v)
| . | . | 37. lnk(v) = l
| . | . | 38. sender(v) = e
| . | . | 39. index(v) = index(e')
| . | . | 40. LL(index(e')) = v
| . | . | ⊢  loc(e') = loc(v)
| . | . |   by InstESAxiom ES(w) [e'] 13 THEN InstESAxiom ES(w) [v] 13
| . | . |      THEN
| . | . |      HypSubst -6 -1
| . | . |      THEN
| . | . |      HypSubst -11 -2
| . | . |
| . | . .\
| . | . . 1. i : Id
| . | . . 2. k : Knd
| . | . . 3. l : IdLnk
| . | . . 4. ds : x:Id fp-> Type
| . | . . 5. da : a:Knd fp-> Type
| . | . . 6. f :
| . | . . |  (tg:Id
| . | . . |   ×State(ds)→ma-valtype(da; k)→
| . | . . |   (da(rcv(l; tg))?Void List)) List
| . | . . 7. source(l) = i
| . | . . 8. D' : Dsys
| . | . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | . . 10. w : World
| . | . . 11. p : FairFifo
| . | . . 12. PossibleWorld(D';w)
| . | . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . . 14. ∀e:E.
| . | . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . . 15. ∀e:E.
| . | . . |   isrcv(e)
| . | . . |   ⇒
| . | . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . | . . 17. ∀e:E.
| . | . . |   loc(e) = i
| . | . . |   ⇒
| . | . . |   kind(e) = k
| . | . . |   ⇒
| . | . . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . . 19. ∀e:E.
| . | . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . . 20. ∀e:E.
| . | . . |   isrcv(e)
| . | . . |   ⇒
```

```
| . | . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . . 21. e : E
| . | . . 22. loc(e) = i
| . | . . 23. kind(e) = k
| . | . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . . 27. L :
| . | . . |   ∀n:ℕ||sends(l;e)||.
| . | . . |   ∃e':E.
| . | . . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | . . 28. LL :
| . | . . |   n:ℕ||sends(l;e)||→{e':E
| . | . . |                         | isrcv(e')
| . | . . |                           & lnk(e') = l
| . | . . |                             & sender(e') = e
| . | . . |                             & index(e') = n }
| . | . . 29. (λn.1of(L(n))) = LL
| . | . . 30. e' : E
| . | . . 31. isrcv(e')
| . | . . 32. lnk(e') = l
| . | . . 33. sender(e') = e
| . | . . 34. index(e') ∈ ℕ||sends(l;e)||
| . | . . 35. v : E
| . | . . 36. isrcv(v)
| . | . . 37. lnk(v) = l
| . | . . 38. sender(v) = e
| . | . . 39. index(v) = index(e')
| . | . . 40. LL(index(e')) = v
| . | . . 41. (e' <loc v) ∨ e' = v ∨ (v <loc e')
| . | . . ⊢  e' = v  by SplitOrHyps
| . | . . |
| . | . . |\
| . | . . | 1. i : Id
| . | . . | 2. k : Knd
| . | . . | 3. l : IdLnk
| . | . . | 4. ds : x:Id fp-> Type
| . | . . | 5. da : a:Knd fp-> Type
| . | . . | 6. f :
| . | . . |    (tg:Id
| . | . . |     ×State(ds)→ma-valtype(da; k)→
| . | . . |     (da(rcv(l; tg))?Void List)) List
| . | . . | 7. source(l) = i
| . | . . | 8. D' : Dsys
| . | . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | . . | 10. w : World
| . | . . | 11. p : FairFifo
| . | . . | 12. PossibleWorld(D';w)
| . | . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . . | 14. ∀e:E.
| . | . . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . . | 15. ∀e:E.
| . | . . |     isrcv(e)
| . | . . |     ⇒
| . | . . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
```

```
| . | . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | . . |     ((λl,tg. da(rcv(l; tg))?Top))
| . | . . | 17. ∀e:E.
| . | . . |     loc(e) = i
| . | . . |     ⇒
| . | . . |     kind(e) = k
| . | . . |     ⇒
| . | . . |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . . | 19. ∀e:E.
| . | . . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . . | 20. ∀e:E.
| . | . . |     isrcv(e)
| . | . . |     ⇒
| . | . . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . . | 21. e : E
| . | . . | 22. loc(e) = i
| . | . . | 23. kind(e) = k
| . | . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . . | 27. L :
| . | . . |     ∀n:ℕ||sends(l;e)||.
| . | . . |     ∃e':E.
| . | . . |     isrcv(e')
| . | . . |     & lnk(e') = l & sender(e') = e & index(e') = n
| . | . . | 28. LL :
| . | . . |     n:ℕ||sends(l;e)||→{e':E
| . | . . |                         | isrcv(e')
| . | . . |                           & lnk(e') = l
| . | . . |                             & sender(e') = e
| . | . . |                             & index(e') = n }
| . | . . | 29. (λn.1of(L(n))) = LL
| . | . . | 30. e' : E
| . | . . | 31. isrcv(e')
| . | . . | 32. lnk(e') = l
| . | . . | 33. sender(e') = e
| . | . . | 34. index(e') ∈ ℕ||sends(l;e)||
| . | . . | 35. v : E
| . | . . | 36. isrcv(v)
| . | . . | 37. lnk(v) = l
| . | . . | 38. sender(v) = e
| . | . . | 39. index(v) = index(e')
| . | . . | 40. LL(index(e')) = v
| . | . . | 41. (e' <loc v)
| . | . . | ⊢  e' = v
| . | . . |   by InstESAxiom ES(w) [e';v] 15 THEN ThinTrivial
| . | . . |      THEN
| . | . . |      SplitOrHyps
| . | . . |      THEN
| . | . . |      Inst
| . | . . |      Thm* ∀the_es:ES, e:E. ¬(e <loc e)
| . | . . |      {es_DASH_locl_DASH_antireflexive}
| . | . . |      [ES(w);e]
| . | . . |      THEN
```

```
| . | . . |      Analyze -1
| . | . . |      THEN
| . | . . |      HypSubst -6 -1
| . | . . |      THEN
| . | . . |      HypSubst -11 -1
| . | . . |
| . | . . .\
| . | . . . 1. i : Id
| . | . . . 2. k : Knd
| . | . . . 3. l : IdLnk
| . | . . . 4. ds : x:Id fp-> Type
| . | . . . 5. da : a:Knd fp-> Type
| . | . . . 6. f :
| . | . . .    (tg:Id
| . | . . .     ×State(ds)→ma-valtype(da; k)→
| . | . . .     (da(rcv(l; tg))?Void List)) List
| . | . . . 7. source(l) = i
| . | . . . 8. D' : Dsys
| . | . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | . . . 10. w : World
| . | . . . 11. p : FairFifo
| . | . . . 12. PossibleWorld(D';w)
| . | . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . . . 14. ∀e:E.
| . | . . .     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . . . 15. ∀e:E.
| . | . . .     isrcv(e)
| . | . . .     ⇒
| . | . . .     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | . . .     ((λl,tg. da(rcv(l; tg))?Top))
| . | . . . 17. ∀e:E.
| . | . . .     loc(e) = i
| . | . . .     ⇒
| . | . . .     kind(e) = k
| . | . . .     ⇒
| . | . . .     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | . . . 19. ∀e:E.
| . | . . .     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . . . 20. ∀e:E.
| . | . . .     isrcv(e)
| . | . . .     ⇒
| . | . . .     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | . . . 21. e : E
| . | . . . 22. loc(e) = i
| . | . . . 23. kind(e) = k
| . | . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . . . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | . . . 27. L :
| . | . . .     ∀n:ℕ||sends(l;e)||.
| . | . . .     ∃e':E.
| . | . . .     isrcv(e')
| . | . . .     & lnk(e') = l & sender(e') = e & index(e') = n
```

```
| . | . . . 28. LL :
| . | . . .     n:ℕ||sends(l;e)||→{e':E
| . | . . .                          | isrcv(e')
| . | . . .                            & lnk(e') = l
| . | . . .                              & sender(e') = e
| . | . . .                              & index(e') = n }
| . | . . . 29. (λn.1of(L(n))) = LL
| . | . . . 30. e' : E
| . | . . . 31. isrcv(e')
| . | . . . 32. lnk(e') = l
| . | . . . 33. sender(e') = e
| . | . . . 34. index(e') ∈ ℕ||sends(l;e)||
| . | . . . 35. v : E
| . | . . . 36. isrcv(v)
| . | . . . 37. lnk(v) = l
| . | . . . 38. sender(v) = e
| . | . . . 39. index(v) = index(e')
| . | . . . 40. LL(index(e')) = v
| . | . . . 41. (v <loc e')
| . | . . . ⊢  e' = v
| . | . . .   by InstESAxiom ES(w) [v;e'] 15 THEN ThinTrivial
| . | . . .      THEN
| . | . . .      SplitOrHyps
| . | . . .      THEN
| . | . . .      Inst
| . | . . .      Thm* ∀the_es:ES, e:E. ¬(e <loc e)
| . | . . .      {es_DASH_locl_DASH_antireflexive}
| . | . . .      [ES(w);e]
| . | . . .      THEN
| . | . . .      Analyze -1
| . | . . .      THEN
| . | . . .      HypSubst -6 -1
| . | . . .      THEN
| . | . . .      HypSubst -11 -1
| . |
| . |\
| . | 1. i : Id
| . | 2. k : Knd
| . | 3. l : IdLnk
| . | 4. ds : x:Id fp-> Type
| . | 5. da : a:Knd fp-> Type
| . | 6. f :
| . | |  (tg:Id
| . | |   ×State(ds)→ma-valtype(da; k)→
| . | |   (da(rcv(l; tg))?Void List)) List
| . | 7. source(l) = i
| . | 8. D' : Dsys
| . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | 10. w : World
| . | 11. p : FairFifo
| . | 12. PossibleWorld(D';w)
| . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 15. ∀e:E.
| . | |   isrcv(e)
```

```
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . | 17. ∀e:E.
| . | |   loc(e) = i
| . | |   ⇒
| . | |   kind(e) = k
| . | |   ⇒
| . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 20. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 21. e : E
| . | 22. loc(e) = i
| . | 23. kind(e) = k
| . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 27. L :
| . | |   ∀n:ℕ||sends(l;e)||.
| . | |   ∃e':E.
| . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | 28. LL :
| . | |   n:ℕ||sends(l;e)||→{e':E
| . | |                      | isrcv(e')
| . | |                        & lnk(e') = l
| . | |                          & sender(e') = e
| . | |                          & index(e') = n }
| . | 29. (λn.1of(L(n))) = LL
| . | ⊢  ∀e1,e2:E.
| . | |  e1 before e2 ∈ map(LL;upto(||sends(l;e)||)) ⇒ (e1 <loc e2)
| . | | by Auto
| . | |
| . | 1. i : Id
| . | 2. k : Knd
| . | 3. l : IdLnk
| . | 4. ds : x:Id fp-> Type
| . | 5. da : a:Knd fp-> Type
| . | 6. f :
| . | |  (tg:Id
| . | |   ×State(ds)→ma-valtype(da; k)→
| . | |  (da(rcv(l; tg))?Void List)) List
| . | 7. source(l) = i
| . | 8. D' : Dsys
| . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | 10. w : World
| . | 11. p : FairFifo
| . | 12. PossibleWorld(D';w)
| . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 15. ∀e:E.
```

```
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . | 17. ∀e:E.
| . | |   loc(e) = i
| . | |   ⇒
| . | |   kind(e) = k
| . | |   ⇒
| . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 20. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 21. e : E
| . | 22. loc(e) = i
| . | 23. kind(e) = k
| . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 27. L :
| . | |   ∀n:ℕ||sends(l;e)||.
| . | |   ∃e':E.
| . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | 28. LL :
| . | |   n:ℕ||sends(l;e)||→{e':E
| . | |                     | isrcv(e')
| . | |                       & lnk(e') = l
| . | |                         & sender(e') = e
| . | |                         & index(e') = n }
| . | 29. (λn.1of(L(n))) = LL
| . | 30. e1 : E
| . | 31. e2 : E
| . | 32. e1 before e2 ∈ map(LL;upto(||sends(l;e)||))
| . | ⊢  (e1 <loc e2)
| . | | by RWO
| . | |    Thm* ∀f:(T→T'), L:T List, x',y':T'.
| . | |         x' before y' ∈ map(f;L)
| . | |         ⟺
| . | |         (∃x,y:T. x before y ∈ L & f(x) = x' & f(y) = y')
| . | |    {before_DASH_map}
| . | |    -1
| . | |    THEN
| . | |    ExRepD
| . | |
| . | 1. i : Id
| . | 2. k : Knd
| . | 3. l : IdLnk
| . | 4. ds : x:Id fp-> Type
| . | 5. da : a:Knd fp-> Type
| . | 6. f :
| . | |  (tg:Id
```

```
| . | |   ×State(ds)→ma-valtype(da; k)→
| . | |   (da(rcv(l; tg))?Void List)) List
| . | 7. source(l) = i
| . | 8. D' : Dsys
| . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | 10. w : World
| . | 11. p : FairFifo
| . | 12. PossibleWorld(D';w)
| . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 15. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . | 17. ∀e:E.
| . | |   loc(e) = i
| . | |   ⇒
| . | |   kind(e) = k
| . | |   ⇒
| . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 20. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 21. e : E
| . | 22. loc(e) = i
| . | 23. kind(e) = k
| . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 27. L :
| . | |   ∀n:ℕ||sends(l;e)||.
| . | |   ∃e':E.
| . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | 28. LL :
| . | |   n:ℕ||sends(l;e)||→{e':E
| . | |                     | isrcv(e')
| . | |                       & lnk(e') = l
| . | |                         & sender(e') = e
| . | |                         & index(e') = n }
| . | 29. (λn.1of(L(n))) = LL
| . | 30. e1 : E
| . | 31. e2 : E
| . | 32. x : ℕ||sends(l;e)||
| . | 33. y : ℕ||sends(l;e)||
| . | 34. x before y ∈ upto(||sends(l;e)||)
| . | 35. LL(x) = e1
| . | 36. LL(y) = e2
| . | ⊢  (e1 <loc e2)  by RevHypSubst -2 0 THEN RevHypSubst -1 0
| . | |
| . | 1. i : Id
```

```
| . | 2. k : Knd
| . | 3. l : IdLnk
| . | 4. ds : x:Id fp-> Type
| . | 5. da : a:Knd fp-> Type
| . | 6. f :
| . | |  (tg:Id
| . | |   ×State(ds)→ma-valtype(da; k)→
| . | |   (da(rcv(l; tg))?Void List)) List
| . | 7. source(l) = i
| . | 8. D' : Dsys
| . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | 10. w : World
| . | 11. p : FairFifo
| . | 12. PossibleWorld(D';w)
| . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 15. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . | 17. ∀e:E.
| . | |   loc(e) = i
| . | |   ⇒
| . | |   kind(e) = k
| . | |   ⇒

| . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 20. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 21. e : E
| . | 22. loc(e) = i
| . | 23. kind(e) = k
| . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 27. L :
| . | |   ∀n:ℕ||sends(l;e)||.
| . | |   ∃e':E.
| . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | 28. LL :
| . | |   n:ℕ||sends(l;e)||→{e':E
| . | |                      | isrcv(e')
| . | |                        & lnk(e') = l
| . | |                          & sender(e') = e
| . | |                          & index(e') = n }
| . | 29. (λn.1of(L(n))) = LL
| . | 30. e1 : E
| . | 31. e2 : E
| . | 32. x : ℕ||sends(l;e)||
| . | 33. y : ℕ||sends(l;e)||
```

```
| . | 34. x before y ∈ upto(||sends(l;e)||)
| . | 35. LL(x) = e1
| . | 36. LL(y) = e2
| . | ⊢  (LL(x) <loc LL(y))
| . | | by GenConclAtAddr [2] THEN Thin -1 THEN GenConclAtAddr [3]
| . | |    THEN
| . | |    Thin -1
| . | |
| . | 1. i : Id
| . | 2. k : Knd
| . | 3. l : IdLnk
| . | 4. ds : x:Id fp-> Type
| . | 5. da : a:Knd fp-> Type
| . | 6. f :
| . | |  (tg:Id
| . | |   ×State(ds)→ma-valtype(da; k)→
| . | |   (da(rcv(l; tg))?Void List)) List
| . | 7. source(l) = i
| . | 8. D' : Dsys
| . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | 10. w : World
| . | 11. p : FairFifo
| . | 12. PossibleWorld(D';w)
| . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 15. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . | 17. ∀e:E.
| . | |   loc(e) = i
| . | |   ⇒
| . | |   kind(e) = k
| . | |   ⇒
| . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 20. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 21. e : E
| . | 22. loc(e) = i
| . | 23. kind(e) = k
| . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 27. L :
| . | |   ∀n:ℕ||sends(l;e)||.
| . | |   ∃e':E.
| . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | 28. LL :
| . | |   n:ℕ||sends(l;e)||→{e':E
```

```
| . | |                       | isrcv(e')
| . | |                         & lnk(e') = l
| . | |                           & sender(e') = e
| . | |                           & index(e') = n }
| . | 29. (λn.1of(L(n))) = LL
| . | 30. e1 : E
| . | 31. e2 : E
| . | 32. x : ℕ||sends(l;e)||
| . | 33. y : ℕ||sends(l;e)||
| . | 34. x before y ∈ upto(||sends(l;e)||)
| . | 35. LL(x) = e1
| . | 36. LL(y) = e2
| . | 37. v :
| . | |   {e':E
| . | |    | isrcv(e')
| . | |      & lnk(e') = l & sender(e') = e & index(e') = x }
| . | 38. v1 :
| . | |   {e':E
| . | |    | isrcv(e')
| . | |      & lnk(e') = l & sender(e') = e & index(e') = y }
| . | ⊢  (v <loc v1)
| . | | by DVar `v' THEN DVar `v1' THEN InstESAxiom ES(w) [v;v1] 15
| . | |    THENA
| . | |    (Auto THEN Unhide THEN ExRepD)
| . | |
| . | 1. i : Id
| . | 2. k : Knd
| . | 3. l : IdLnk
| . | 4. ds : x:Id fp-> Type
| . | 5. da : a:Knd fp-> Type
| . | 6. f :
| . | |  (tg:Id
| . | |   ×State(ds)→ma-valtype(da; k)→
| . | |   (da(rcv(l; tg))?Void List)) List
| . | 7. source(l) = i
| . | 8. D' : Dsys
| . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | 10. w : World
| . | 11. p : FairFifo
| . | 12. PossibleWorld(D';w)
| . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 15. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . | 17. ∀e:E.
| . | |   loc(e) = i
| . | |   ⇒
| . | |   kind(e) = k
| . | |   ⇒
| . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
```

```
| . | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 20. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 21. e : E
| . | 22. loc(e) = i
| . | 23. kind(e) = k
| . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 27. L :
| . | |   ∀n:ℕ||sends(l;e)||.
| . | |   ∃e':E.
| . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | 28. LL :
| . | |   n:ℕ||sends(l;e)||→{e':E
| . | |                       | isrcv(e')
| . | |                         & lnk(e') = l
| . | |                           & sender(e') = e
| . | |                           & index(e') = n }
| . | 29. (λn.1of(L(n))) = LL
| . | 30. e1 : E
| . | 31. e2 : E
| . | 32. x : ℕ||sends(l;e)||
| . | 33. y : ℕ||sends(l;e)||
| . | 34. x before y ∈ upto(||sends(l;e)||)
| . | 35. LL(x) = e1
| . | 36. LL(y) = e2
| . | 37. v : E
| . | 38. isrcv(v)
| . | |   & lnk(v) = l
| . | |     & sender(v) = e
| . | |     & index(v) = x  [not for witness]
| . | 39. v1 : E
| . | 40. isrcv(v1)
| . | |   & lnk(v1) = l
| . | |     & sender(v1) = e
| . | |     & index(v1) = y  [not for witness]
| . | 41. (v <loc v1)
| . | |   ⟺
| . | |   (sender(v) <loc sender(v1))
| . | |   ∨ sender(v) = sender(v1) & index(v)<index(v1)
| . | ⊢  (v <loc v1)  by BackThruSomeHyp
| . | |
| . | 1. i : Id
| . | 2. k : Knd
| . | 3. l : IdLnk
| . | 4. ds : x:Id fp-> Type
| . | 5. da : a:Knd fp-> Type
| . | 6. f :
| . | |  (tg:Id
| . | |   ×State(ds)→ma-valtype(da; k)→
| . | |   (da(rcv(l; tg))?Void List)) List
| . | 7. source(l) = i
```

```
| . | 8. D' : Dsys
| . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | 10. w : World
| . | 11. p : FairFifo
| . | 12. PossibleWorld(D';w)
| . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 15. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . | 17. ∀e:E.
| . | |   loc(e) = i
| . | |   ⇒
| . | |   kind(e) = k
| . | |   ⇒
| . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 20. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 21. e : E
| . | 22. loc(e) = i
| . | 23. kind(e) = k
| . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 27. L :
| . | |   ∀n:ℕ||sends(l;e)||.
| . | |   ∃e':E.
| . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | 28. LL :
| . | |   n:ℕ||sends(l;e)||→{e':E
| . | |                      | isrcv(e')
| . | |                        & lnk(e') = l
| . | |                          & sender(e') = e
| . | |                          & index(e') = n }
| . | 29. (λn.1of(L(n))) = LL
| . | 30. e1 : E
| . | 31. e2 : E
| . | 32. x : ℕ||sends(l;e)||
| . | 33. y : ℕ||sends(l;e)||
| . | 34. x before y ∈ upto(||sends(l;e)||)
| . | 35. LL(x) = e1
| . | 36. LL(y) = e2
| . | 37. v : E
| . | 38. isrcv(v)
| . | |   & lnk(v) = l
| . | |     & sender(v) = e
| . | |     & index(v) = x  [not for witness]
| . | 39. v1 : E
```

```
| . | 40. isrcv(v1)
| . | |   & lnk(v1) = l
| . | |     & sender(v1) = e
| . | |     & index(v1) = y  [not for witness]
| . | 41. (v <loc v1)
| . | |   ⟺
| . | |   (sender(v) <loc sender(v1))
| . | |   ∨ sender(v) = sender(v1) & index(v)<index(v1)
| . | ⊢  (sender(v) <loc sender(v1))
| . | |  ∨ sender(v) = sender(v1) & index(v)<index(v1)
| . | | by OrRight THEN ExRepD
| . | |
| . | 1. i : Id
| . | 2. k : Knd
| . | 3. l : IdLnk
| . | 4. ds : x:Id fp-> Type
| . | 5. da : a:Knd fp-> Type
| . | 6. f :
| . | |  (tg:Id
| . | |   ×State(ds)→ma-valtype(da; k)→
| . | |   (da(rcv(l; tg))?Void List)) List
| . | 7. source(l) = i
| . | 8. D' : Dsys
| . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | 10. w : World
| . | 11. p : FairFifo
| . | 12. PossibleWorld(D';w)
| . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 15. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . | 17. ∀e:E.
| . | |   loc(e) = i
| . | |   ⇒
| . | |   kind(e) = k
| . | |   ⇒
| . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 20. ∀e:E.
| . | |   isrcv(e)
| . | |   ⇒
| . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 21. e : E
| . | 22. loc(e) = i
| . | 23. kind(e) = k
| . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)

| . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 27. L :
| . | |   ∀n:ℕ||sends(l;e)||.
```

```
| . | |   ∃e':E.
| . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | 28. LL :
| . | |   n:ℕ||sends(l;e)||→{e':E
| . | |                       | isrcv(e')
| . | |                         & lnk(e') = l
| . | |                           & sender(e') = e
| . | |                           & index(e') = n }
| . | 29. (λn.1of(L(n))) = LL
| . | 30. e1 : E
| . | 31. e2 : E
| . | 32. x : ℕ||sends(l;e)||
| . | 33. y : ℕ||sends(l;e)||
| . | 34. x before y ∈ upto(||sends(l;e)||)
| . | 35. LL(x) = e1
| . | 36. LL(y) = e2
| . | 37. v : E
| . | 38. isrcv(v)
| . | 39. lnk(v) = l
| . | 40. sender(v) = e
| . | 41. index(v) = x
| . | 42. v1 : E
| . | 43. isrcv(v1)
| . | 44. lnk(v1) = l
| . | 45. sender(v1) = e
| . | 46. index(v1) = y
| . | 47. (v <loc v1)
| . | |   ⇒
| . | |   (sender(v) <loc sender(v1))
| . | |   ∨ sender(v) = sender(v1) & index(v)<index(v1)
| . | 48. (v <loc v1) ⇐ ((sender(v) <loc sender(v1))
| . | |   ∨ sender(v) = sender(v1) & index(v)<index(v1))
| . | ⊢  index(v)<index(v1)  by Assert (x<y)
| . | |
| . | .....assertion .....
| . | 1. i : Id
| . | 2. k : Knd
| . | 3. l : IdLnk
| . | 4. ds : x:Id fp-> Type
| . | 5. da : a:Knd fp-> Type
| . | 6. f :
| . |    (tg:Id
| . |     ×State(ds)→ma-valtype(da; k)→
| . |     (da(rcv(l; tg))?Void List)) List
| . | 7. source(l) = i
| . | 8. D' : Dsys
| . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . | 10. w : World
| . | 11. p : FairFifo
| . | 12. PossibleWorld(D';w)
| . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 15. ∀e:E.
| . |     isrcv(e)
| . |     ⇒
```

```
| . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . |     ((λl,tg. da(rcv(l; tg))?Top))
| . | 17. ∀e:E.
| . |     loc(e) = i
| . |     ⇒
| . |     kind(e) = k
| . |     ⇒
| . |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 20. ∀e:E.
| . |     isrcv(e)
| . |     ⇒
| . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 21. e : E
| . | 22. loc(e) = i
| . | 23. kind(e) = k
| . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . | 27. L :
| . |     ∀n:ℕ||sends(l;e)||.
| . |     ∃e':E.
| . |     isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . | 28. LL :
| . |     n:ℕ||sends(l;e)||→{e':E
| . |                          | isrcv(e')
| . |                            & lnk(e') = l
| . |                              & sender(e') = e
| . |                              & index(e') = n }
| . | 29. (λn.1of(L(n))) = LL
| . | 30. e1 : E
| . | 31. e2 : E
| . | 32. x : ℕ||sends(l;e)||
| . | 33. y : ℕ||sends(l;e)||
| . | 34. x before y ∈ upto(||sends(l;e)||)
| . | 35. LL(x) = e1
| . | 36. LL(y) = e2
| . | 37. v : E
| . | 38. isrcv(v)
| . | 39. lnk(v) = l
| . | 40. sender(v) = e
| . | 41. index(v) = x
| . | 42. v1 : E
| . | 43. isrcv(v1)
| . | 44. lnk(v1) = l
| . | 45. sender(v1) = e
| . | 46. index(v1) = y
| . | 47. (v <loc v1)
| . |     ⇒
| . |     (sender(v) <loc sender(v1))
| . |     ∨ sender(v) = sender(v1) & index(v)<index(v1)
| . | 48. (v <loc v1) ⇐ ((sender(v) <loc sender(v1))
| . |     ∨ sender(v) = sender(v1) & index(v)<index(v1))
```

```
| . | ⊢  x<y
| . |   by AllHyps
| . |      (λh.
| . |       RWO
| . |       Thm* ∀n:ℕ, x,y:ℕn. x before y ∈ upto(n) ⟺ x<y
| . |       {before_DASH_upto}
| . |       h)
| . |
| . .\
| . . 1. i : Id
| . . 2. k : Knd
| . . 3. l : IdLnk
| . . 4. ds : x:Id fp-> Type
| . . 5. da : a:Knd fp-> Type
| . . 6. f :
| . . |  (tg:Id
| . . |   ×State(ds)→ma-valtype(da; k)→
| . . |   (da(rcv(l; tg))?Void List)) List
| . . 7. source(l) = i
| . . 8. D' : Dsys
| . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . 10. w : World
| . . 11. p : FairFifo
| . . 12. PossibleWorld(D';w)
| . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . 15. ∀e:E.
| . . |   isrcv(e)
| . . |   ⇒
| . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . 17. ∀e:E.
| . . |   loc(e) = i
| . . |   ⇒
| . . |   kind(e) = k
| . . |   ⇒
| . . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . 20. ∀e:E.
| . . |   isrcv(e)
| . . |   ⇒
| . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . 21. e : E
| . . 22. loc(e) = i
| . . 23. kind(e) = k
| . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . 27. L :
| . . |   ∀n:ℕ||sends(l;e)||.
| . . |   ∃e':E.
| . . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . 28. LL :
```

```
| . . |   n:ℕ||sends(l;e)||→{e':E
| . . |                        | isrcv(e')
| . . |                          & lnk(e') = l
| . . |                            & sender(e') = e
| . . |                            & index(e') = n }
| . . 29. (λn.1of(L(n))) = LL
| . . ⊢  map(λe'.<tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))
| . . |  =
| . . |  tagged-list-messages(λz.(z when e);val(e);f)
| . . | by Subst
| . . |    (tagged-list-messages(λz.(z when e);val(e);f)
| . . |     =
| . . |     map(λm.2of(m);sends(l;e)))
| . . |    0
| . . |
| . . |\
| . . | .....equality .....
| . . | 1. i : Id
| . . | 2. k : Knd
| . . | 3. l : IdLnk
| . . | 4. ds : x:Id fp-> Type
| . . | 5. da : a:Knd fp-> Type
| . . | 6. f :
| . . | |  (tg:Id
| . . | |   ×State(ds)→ma-valtype(da; k)→
| . . | |   (da(rcv(l; tg))?Void List)) List
| . . | 7. source(l) = i
| . . | 8. D' : Dsys
| . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | 10. w : World
| . . | 11. p : FairFifo
| . . | 12. PossibleWorld(D';w)
| . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | 14. ∀e:E.
| . . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | 15. ∀e:E.
| . . | |   isrcv(e)
| . . | |   ⇒
| . . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | 17. ∀e:E.
| . . | |   loc(e) = i
| . . | |   ⇒
| . . | |   kind(e) = k
| . . | |   ⇒
| . . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | 19. ∀e:E.
| . . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | 20. ∀e:E.
| . . | |   isrcv(e)
| . . | |   ⇒
| . . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | 21. e : E
```

```
| . . | 22. loc(e) = i
| . . | 23. kind(e) = k
| . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | 27. L :
| . . | |   ∀n:ℕ||sends(l;e)||.
| . . | |   ∃e':E.
| . . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | 28. LL :
| . . | |   n:ℕ||sends(l;e)||→{e':E
| . . | |                         | isrcv(e')
| . . | |                           & lnk(e') = l
| . . | |                             & sender(e') = e
| . . | |                             & index(e') = n }
| . . | 29. (λn.1of(L(n))) = LL
| . . | ⊢  tagged-list-messages(λz.(z when e);val(e);f)
| . . | |  =
| . . | |  map(λm.2of(m);sends(l;e))
| . . | | by AssertBY (val(e) ∈ ma-valtype(da; k))
| . . | |    (DoSubsume THEN RevHypSubst' -7 0 THEN BackThruSomeHyp)
| . . | |    THEN
| . . | |    Assert ((λz.(z when e)) ∈ State(ds))
| . . | |
| . . | |\
| . . | | .....assertion .....
| . . | | 1. i : Id
| . . | | 2. k : Knd
| . . | | 3. l : IdLnk
| . . | | 4. ds : x:Id fp-> Type
| . . | | 5. da : a:Knd fp-> Type
| . . | | 6. f :
| . . | |    (tg:Id
| . . | |     ×State(ds)→ma-valtype(da; k)→
| . . | |     (da(rcv(l; tg))?Void List)) List
| . . | | 7. source(l) = i
| . . | | 8. D' : Dsys
| . . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | 10. w : World
| . . | | 11. p : FairFifo
| . . | | 12. PossibleWorld(D';w)
| . . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | 14. ∀e:E.
| . . | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 15. ∀e:E.
| . . | |     isrcv(e)
| . . | |     ⇒
| . . | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | | 17. ∀e:E.
| . . | |     loc(e) = i
| . . | |     ⇒
| . . | |     kind(e) = k
| . . | |     ⇒
```

```
| . . | |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | 19. ∀e:E.
| . . | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 20. ∀e:E.
| . . | |     isrcv(e)
| . . | |     ⇒
| . . | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 21. e : E
| . . | | 22. loc(e) = i
| . . | | 23. kind(e) = k
| . . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | 27. L :
| . . | |     ∀n:ℕ||sends(l;e)||.
| . . | |     ∃e':E.
| . . | |     isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | 28. LL :
| . . | |     n:ℕ||sends(l;e)||→{e':E
| . . | |                         | isrcv(e')
| . . | |                           & lnk(e') = l
| . . | |                             & sender(e') = e
| . . | |                             & index(e') = n }
| . . | | 29. (λn.1of(L(n))) = LL
| . . | | 30. val(e) ∈ ma-valtype(da; k)
| . . | | ⊢  (λz.(z when e)) ∈ State(ds)
| . . | |   by Unfold `ma-state` 0 THEN DoSubsume
| . . | |      THEN
| . . | |      Subst (loc(e) = i) 0
| . . | |      THEN
| . . | |      BackThruSomeHyp
| . . | |
| . . | .\
| . . | . 1. i : Id
| . . | . 2. k : Knd
| . . | . 3. l : IdLnk
| . . | . 4. ds : x:Id fp-> Type
| . . | . 5. da : a:Knd fp-> Type
| . . | . 6. f :
| . . | . |  (tg:Id
| . . | . |   ×State(ds)→ma-valtype(da; k)→
| . . | . |   (da(rcv(l; tg))?Void List)) List
| . . | . 7. source(l) = i
| . . | . 8. D' : Dsys
| . . | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . 10. w : World
| . . | . 11. p : FairFifo
| . . | . 12. PossibleWorld(D';w)
| . . | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . 14. ∀e:E.
| . . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . 15. ∀e:E.
| . . | . |   isrcv(e)
| . . | . |   ⇒
```

```
| . . | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . 17. ∀e:E.
| . . | . |   loc(e) = i
| . . | . |   ⇒
| . . | . |   kind(e) = k
| . . | . |   ⇒
| . . | . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . 19. ∀e:E.
| . . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . 20. ∀e:E.
| . . | . |   isrcv(e)
| . . | . |   ⇒
| . . | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . 21. e : E
| . . | . 22. loc(e) = i
| . . | . 23. kind(e) = k
| . . | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . 27. L :
| . . | . |   ∀n:ℕ||sends(l;e)||.
| . . | . |   ∃e':E.
| . . | . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . 28. LL :
| . . | . |   n:ℕ||sends(l;e)||→{e':E
| . . | . |                       | isrcv(e')
| . . | . |                         & lnk(e') = l
| . . | . |                           & sender(e') = e
| . . | . |                           & index(e') = n }
| . . | . 29. (λn.1of(L(n))) = LL
| . . | . 30. val(e) ∈ ma-valtype(da; k)
| . . | . 31. (λz.(z when e)) ∈ State(ds)
| . . | . ⊢  tagged-list-messages(λz.(z when e);val(e);f)
| . . | . |  =
| . . | . |  map(λm.2of(m);sends(l;e))
| . . | . | by Subst
| . . | . |    (sends(l;e)
| . . | . |     =
| . . | . |     tagged-messages(l;λz.(z when e);val(e);f))
| . . | . |    0
| . . | . |
| . . | . |\
| . . | . | .....equality .....
| . . | . | 1. i : Id
| . . | . | 2. k : Knd
| . . | . | 3. l : IdLnk
| . . | . | 4. ds : x:Id fp-> Type
| . . | . | 5. da : a:Knd fp-> Type
| . . | . | 6. f :
| . . | . | |  (tg:Id
| . . | . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . | |   (da(rcv(l; tg))?Void List)) List
```

```
| . . | . | 7. source(l) = i
| . . | . | 8. D' : Dsys
| . . | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . | 10. w : World
| . . | . | 11. p : FairFifo
| . . | . | 12. PossibleWorld(D';w)
| . . | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 14. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 15. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . | 17. ∀e:E.
| . . | . | |   loc(e) = i
| . . | . | |   ⇒
| . . | . | |   kind(e) = k
| . . | . | |   ⇒
| . . | . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 19. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 20. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 21. e : E
| . . | . | 22. loc(e) = i
| . . | . | 23. kind(e) = k
| . . | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 27. L :
| . . | . | |   ∀n:ℕ||sends(l;e)||.
| . . | . | |   ∃e':E.
| . . | . | |   isrcv(e')
| . . | . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | 28. LL :
| . . | . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . | |                      | isrcv(e')
| . . | . | |                        & lnk(e') = l
| . . | . | |                          & sender(e') = e
| . . | . | |                          & index(e') = n }
| . . | . | 29. (λn.1of(L(n))) = LL
| . . | . | 30. val(e) ∈ ma-valtype(da; k)
| . . | . | 31. (λz.(z when e)) ∈ State(ds)
| . . | . | ⊢  sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | | by Id
| . . | . | |
| . . | . | .....equality .....
| . . | . | 1. i : Id
| . . | . | 2. k : Knd
| . . | . | 3. l : IdLnk
```

```
| . . | . | 4. ds : x:Id fp-> Type
| . . | . | 5. da : a:Knd fp-> Type
| . . | . | 6. f :
| . . | . | |  (tg:Id
| . . | . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . | |   (da(rcv(l; tg))?Void List)) List
| . . | . | 7. source(l) = i
| . . | . | 8. D' : Dsys
| . . | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . | 10. w : World
| . . | . | 11. p : FairFifo
| . . | . | 12. PossibleWorld(D';w)
| . . | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 14. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 15. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . | 17. ∀e:E.
| . . | . | |   loc(e) = i
| . . | . | |   ⇒
| . . | . | |   kind(e) = k
| . . | . | |   ⇒
| . . | . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 19. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 20. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 21. e : E
| . . | . | 22. loc(e) = i
| . . | . | 23. kind(e) = k
| . . | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 27. L :
| . . | . | |   ∀n:ℕ||sends(l;e)||.
| . . | . | |   ∃e':E.
| . . | . | |   isrcv(e')
| . . | . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | 28. LL :
| . . | . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . | |                      | isrcv(e')
| . . | . | |                        & lnk(e') = l
| . . | . | |                          & sender(e') = e
| . . | . | |                          & index(e') = n }
| . . | . | 29. (λn.1of(L(n))) = LL
| . . | . | 30. val(e) ∈ ma-valtype(da; k)
| . . | . | 31. (λz.(z when e)) ∈ State(ds)
| . . | . | ⊢  sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
```

```
| . . | . | | by BackThru
| . . | . | |    Thm* ∀P:(T→Prop), A,B:T List.
| . . | . | |         A = B
| . . | . | |         ⇒
| . . | . | |         (∀i:ℕ||A||. P(A[i]))
| . . | . | |         ⇒
| . . | . | |         A = B ∈ {x:T| P(x) } List
| . . | . | |    {list_DASH_eq_DASH_set_DASH_type}
| . . | . | |
| . . | . | 1. i : Id
| . . | . | 2. k : Knd
| . . | . | 3. l : IdLnk
| . . | . | 4. ds : x:Id fp-> Type
| . . | . | 5. da : a:Knd fp-> Type
| . . | . | 6. f :
| . . | . | |  (tg:Id
| . . | . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . | |   (da(rcv(l; tg))?Void List)) List
| . . | . | 7. source(l) = i
| . . | . | 8. D' : Dsys
| . . | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . | 10. w : World
| . . | . | 11. p : FairFifo
| . . | . | 12. PossibleWorld(D';w)
| . . | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 14. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 15. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . | 17. ∀e:E.
| . . | . | |   loc(e) = i
| . . | . | |   ⇒
| . . | . | |   kind(e) = k
| . . | . | |   ⇒
| . . | . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 19. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 20. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 21. e : E
| . . | . | 22. loc(e) = i
| . . | . | 23. kind(e) = k
| . . | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 27. L :
| . . | . | |   ∀n:ℕ||sends(l;e)||.
| . . | . | |   ∃e':E.
```

```
| . . | . | |   isrcv(e')
| . . | . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | 28. LL :
| . . | . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . | |                         | isrcv(e')
| . . | . | |                           & lnk(e') = l
| . . | . | |                             & sender(e') = e
| . . | . | |                             & index(e') = n }
| . . | . | 29. (λn.1of(L(n))) = LL
| . . | . | 30. val(e) ∈ ma-valtype(da; k)
| . . | . | 31. (λz.(z when e)) ∈ State(ds)
| . . | . | 32. i1 : ℕ||sends(l;e)||
| . . | . | ⊢  mlnk(sends(l;e)[i1]) = l  by GenConclAtAddr [2;1]
| . . | . | |
| . . | . | 1. i : Id
| . . | . | 2. k : Knd
| . . | . | 3. l : IdLnk
| . . | . | 4. ds : x:Id fp-> Type
| . . | . | 5. da : a:Knd fp-> Type
| . . | . | 6. f :
| . . | .      (tg:Id
| . . | .       ×State(ds)→ma-valtype(da; k)→
| . . | .       (da(rcv(l; tg))?Void List)) List
| . . | . | 7. source(l) = i
| . . | . | 8. D' : Dsys
| . . | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . | 10. w : World
| . . | . | 11. p : FairFifo
| . . | . | 12. PossibleWorld(D';w)
| . . | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 14. ∀e:E.
| . . | .       loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 15. ∀e:E.
| . . | .       isrcv(e)
| . . | .       ⇒
| . . | .       lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | .       ((λl,tg. da(rcv(l; tg))?Top))
| . . | . | 17. ∀e:E.
| . . | .       loc(e) = i
| . . | .       ⇒
| . . | .       kind(e) = k
| . . | .       ⇒
| . . | .       sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 19. ∀e:E.
| . . | .       loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))

| . . | . | 20. ∀e:E.
| . . | .       isrcv(e)
| . . | .       ⇒
| . . | .       lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 21. e : E
| . . | . | 22. loc(e) = i
| . . | . | 23. kind(e) = k
| . . | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
```

```
| . . | . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 27. L :
| . . | . |     ∀n:ℕ||sends(l;e)||.
| . . | . |     ∃e':E.
| . . | . |     isrcv(e')
| . . | . |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | 28. LL :
| . . | . |     n:ℕ||sends(l;e)||→{e':E
| . . | . |                          | isrcv(e')
| . . | . |                            & lnk(e') = l
| . . | . |                              & sender(e') = e
| . . | . |                              & index(e') = n }
| . . | . | 29. (λn.1of(L(n))) = LL
| . . | . | 30. val(e) ∈ ma-valtype(da; k)
| . . | . | 31. (λz.(z when e)) ∈ State(ds)
| . . | . | 32. i1 : ℕ||sends(l;e)||
| . . | . | 33. v : (Msg on l)
| . . | . | 34. sends(l;e)[i1] = v
| . . | . | ⊢  mlnk(v) = l  by Analyze -2 THEN Unfold `haslink` -2
| . . | . |
| . . | . |\
| . . | . | 1. i : Id
| . . | . | 2. k : Knd
| . . | . | 3. l : IdLnk
| . . | . | 4. ds : x:Id fp-> Type
| . . | . | 5. da : a:Knd fp-> Type
| . . | . | 6. f :
| . . | . | |  (tg:Id
| . . | . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . | |   (da(rcv(l; tg))?Void List)) List
| . . | . | 7. source(l) = i
| . . | . | 8. D' : Dsys
| . . | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . | 10. w : World
| . . | . | 11. p : FairFifo
| . . | . | 12. PossibleWorld(D';w)
| . . | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 14. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 15. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . | 17. ∀e:E.
| . . | . | |   loc(e) = i
| . . | . | |   ⇒
| . . | . | |   kind(e) = k
| . . | . | |   ⇒
| . . | . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 19. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
```

```
| . . | . | 20. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 21. e : E
| . . | . | 22. loc(e) = i
| . . | . | 23. kind(e) = k
| . . | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 27. L :
| . . | . | |   ∀n:ℕ||sends(l;e)||.
| . . | . | |   ∃e':E.
| . . | . | |   isrcv(e')
| . . | . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | 28. LL :
| . . | . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . | |                         | isrcv(e')
| . . | . | |                           & lnk(e') = l
| . . | . | |                             & sender(e') = e
| . . | . | |                             & index(e') = n }
| . . | . | 29. (λn.1of(L(n))) = LL
| . . | . | 30. val(e) ∈ ma-valtype(da; k)
| . . | . | 31. (λz.(z when e)) ∈ State(ds)
| . . | . | ⊢  tagged-list-messages(λz.(z when e);val(e);f)
| . . | . | |  =
| . . | . | |  map(λm.2of(m);tagged-messages(l;λz.
| . . | . | |  (z when e);val(e);f))
| . . | . | | by Unfold `tagged-messages` 0
| . . | . | |    THEN
| . . | . | |    RWO
| . . | . | |    Thm* ∀as:Top List, f,g:Top.
| . . | . | |         map(g;map(f;as)) ~ map(g o f;as)
| . . | . | |    {map_DASH_map}
| . . | . | |    0
| . . | . | |
| . . | . | 1. i : Id
| . . | . | 2. k : Knd
| . . | . | 3. l : IdLnk
| . . | . | 4. ds : x:Id fp-> Type
| . . | . | 5. da : a:Knd fp-> Type
| . . | . | 6. f :
| . . | . | |  (tg:Id
| . . | . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . | |   (da(rcv(l; tg))?Void List)) List
| . . | . | 7. source(l) = i
| . . | . | 8. D' : Dsys
| . . | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . | 10. w : World
| . . | . | 11. p : FairFifo
| . . | . | 12. PossibleWorld(D';w)
| . . | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 14. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 15. ∀e:E.
```

```
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . | 17. ∀e:E.
| . . | . | |   loc(e) = i
| . . | . | |   ⇒
| . . | . | |   kind(e) = k
| . . | . | |   ⇒
| . . | . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 19. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 20. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 21. e : E
| . . | . | 22. loc(e) = i
| . . | . | 23. kind(e) = k
| . . | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 27. L :
| . . | . | |   ∀n:ℕ||sends(l;e)||.
| . . | . | |   ∃e':E.
| . . | . | |   isrcv(e')
| . . | . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | 28. LL :
| . . | . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . | |                     | isrcv(e')
| . . | . | |                       & lnk(e') = l
| . . | . | |                         & sender(e') = e
| . . | . | |                         & index(e') = n }
| . . | . | 29. (λn.1of(L(n))) = LL
| . . | . | 30. val(e) ∈ ma-valtype(da; k)
| . . | . | 31. (λz.(z when e)) ∈ State(ds)
| . . | . | ⊢  tagged-list-messages(λz.(z when e);val(e);f)
| . . | . | |  =
| . . | . | |  map((λm.2of(m)) o (λx.<l,x>);tagged-list-messages(λz.
| . . | . | |  (z when e);val(e);f))
| . . | . | | by Unfold `compose` 0 THEN Reduce 0
| . . | . | |
| . . | . | 1. i : Id
| . . | . | 2. k : Knd
| . . | . | 3. l : IdLnk
| . . | . | 4. ds : x:Id fp-> Type
| . . | . | 5. da : a:Knd fp-> Type
| . . | . | 6. f :
| . . | . | |  (tg:Id
| . . | . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . | |   (da(rcv(l; tg))?Void List)) List
| . . | . | 7. source(l) = i
| . . | . | 8. D' : Dsys
```

```
| . . | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . | 10. w : World
| . . | . | 11. p : FairFifo
| . . | . | 12. PossibleWorld(D';w)
| . . | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 14. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 15. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . | 17. ∀e:E.
| . . | . | |   loc(e) = i
| . . | . | |   ⇒
| . . | . | |   kind(e) = k
| . . | . | |   ⇒
| . . | . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 19. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 20. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 21. e : E

| . . | . | 22. loc(e) = i
| . . | . | 23. kind(e) = k
| . . | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 27. L :
| . . | . | |   ∀n:ℕ||sends(l;e)||.
| . . | . | |   ∃e':E.
| . . | . | |   isrcv(e')
| . . | . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | 28. LL :
| . . | . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . | |                        | isrcv(e')
| . . | . | |                          & lnk(e') = l
| . . | . | |                            & sender(e') = e
| . . | . | |                            & index(e') = n }
| . . | . | 29. (λn.1of(L(n))) = LL
| . . | . | 30. val(e) ∈ ma-valtype(da; k)
| . . | . | 31. (λz.(z when e)) ∈ State(ds)
| . . | . | ⊢  tagged-list-messages(λz.(z when e);val(e);f)
| . . | . | |  =
| . . | . | |  map(λx.x;tagged-list-messages(λz.(z when e);val(e);f))
| . . | . | | by GenConclAtAddr [2]
| . . | . | |
| . . | . | 1. i : Id
| . . | . | 2. k : Knd
| . . | . | 3. l : IdLnk
| . . | . | 4. ds : x:Id fp-> Type
```

```
| . . | . | 5. da : a:Knd fp-> Type
| . . | . | 6. f :
| . . | . |    (tg:Id
| . . | . |     ×State(ds)→ma-valtype(da; k)→
| . . | . |     (da(rcv(l; tg))?Void List)) List
| . . | . | 7. source(l) = i
| . . | . | 8. D' : Dsys
| . . | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . | 10. w : World
| . . | . | 11. p : FairFifo
| . . | . | 12. PossibleWorld(D';w)
| . . | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 14. ∀e:E.
| . . | . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 15. ∀e:E.
| . . | . |     isrcv(e)
| . . | . |     ⇒
| . . | . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . | 17. ∀e:E.
| . . | . |     loc(e) = i
| . . | . |     ⇒
| . . | . |     kind(e) = k
| . . | . |     ⇒
| . . | . |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 19. ∀e:E.
| . . | . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 20. ∀e:E.
| . . | . |     isrcv(e)
| . . | . |     ⇒
| . . | . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 21. e : E
| . . | . | 22. loc(e) = i
| . . | . | 23. kind(e) = k
| . . | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 27. L :
| . . | . |     ∀n:ℕ||sends(l;e)||.
| . . | . |     ∃e':E.
| . . | . |     isrcv(e')
| . . | . |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | 28. LL :
| . . | . |     n:ℕ||sends(l;e)||→{e':E
| . . | . |                         | isrcv(e')
| . . | . |                           & lnk(e') = l
| . . | . |                             & sender(e') = e
| . . | . |                             & index(e') = n }
| . . | . | 29. (λn.1of(L(n))) = LL
| . . | . | 30. val(e) ∈ ma-valtype(da; k)
| . . | . | 31. (λz.(z when e)) ∈ State(ds)
| . . | . | 32. v : (t:Id×da(rcv(l; t))?Void) List
| . . | . | 33. tagged-list-messages(λz.(z when e);val(e);f) = v
```

```
| . . | . | ⊢  v = map(λx.x;v)
| . . | . |   by Lemmaize [] THEN InductionOnList THEN Reduce 0
| . . | . |      THEN
| . . | . |      Analyze
| . . | . |      THEN
| . . | . |      Unfold ‘ma-valtype‘ 0
| . . | . |      THEN
| . . | . |      Analyze 0
| . . | . |      THEN
| . . | . |      TrySubsume
| . . | . |
| . . | . .\
| . . | . . .....wf .....
| . . | . . 1. i : Id
| . . | . . 2. k : Knd
| . . | . . 3. l : IdLnk
| . . | . . 4. ds : x:Id fp-> Type
| . . | . . 5. da : a:Knd fp-> Type
| . . | . . 6. f :
| . . | . . |  (tg:Id
| . . | . . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . |   (da(rcv(l; tg))?Void List)) List
| . . | . . 7. source(l) = i
| . . | . . 8. D’ : Dsys
| . . | . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D’
| . . | . . 10. w : World
| . . | . . 11. p : FairFifo
| . . | . . 12. PossibleWorld(D’;w)
| . . | . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . 14. ∀e:E.
| . . | . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . 15. ∀e:E.
| . . | . . |   isrcv(e)
| . . | . . |   ⇒
| . . | . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . 17. ∀e:E.
| . . | . . |   loc(e) = i
| . . | . . |   ⇒
| . . | . . |   kind(e) = k
| . . | . . |   ⇒
| . . | . . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . 19. ∀e:E.
| . . | . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . 20. ∀e:E.
| . . | . . |   isrcv(e)
| . . | . . |   ⇒
| . . | . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . 21. e : E
| . . | . . 22. loc(e) = i
| . . | . . 23. kind(e) = k
| . . | . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
```

```
| . . | . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . 27. L :
| . . | . . |   ∀n:ℕ||sends(l;e)||.
| . . | . . |   ∃e':E.
| . . | . . |   isrcv(e')
| . . | . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . 28. LL :
| . . | . . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . |                       | isrcv(e')
| . . | . . |                         & lnk(e') = l
| . . | . . |                           & sender(e') = e
| . . | . . |                           & index(e') = n }
| . . | . . 29. (λn.1of(L(n))) = LL
| . . | . . 30. val(e) ∈ ma-valtype(da; k)
| . . | . . 31. (λz.(z when e)) ∈ State(ds)
| . . | . . 32. z :
| . . | . . |   {m:Msg((λl,tg. da(rcv(l; tg))?Top))
| . . | . . |    | mlnk(m) = l } List
| . . | . . ⊢  (tagged-list-messages(λz.(z when e);val(e);f)
| . . | . . |   =
| . . | . . |   map(λm.2of(m);z))
| . . | . . |  ∈ Prop  by Auto
| . . | . . |
| . . | . . |\
| . . | . . | .....unproved Inclusion subgoal .....
| . . | . . | 1. i : Id
| . . | . . | 2. k : Knd
| . . | . . | 3. l : IdLnk
| . . | . . | 4. ds : x:Id fp-> Type
| . . | . . | 5. da : a:Knd fp-> Type
| . . | . . | 6. f :
| . . | . . |    (tg:Id
| . . | . . |     ×State(ds)→ma-valtype(da; k)→
| . . | . . |     (da(rcv(l; tg))?Void List)) List
| . . | . . | 7. source(l) = i
| . . | . . | 8. D' : Dsys
| . . | . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . | 10. w : World
| . . | . . | 11. p : FairFifo
| . . | . . | 12. PossibleWorld(D';w)
| . . | . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . | 14. ∀e:E.
| . . | . . |     loc(e) = i
| . . | . . |     ⇒
| . . | . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . | 15. ∀e:E.
| . . | . . |     isrcv(e)
| . . | . . |     ⇒
| . . | . . |     lnk(e) = l
| . . | . . |     ⇒
| . . | . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . | 17. ∀e:E.
| . . | . . |     loc(e) = i
```

```
| . . | . . |     ⇒
| . . | . . |     kind(e) = k
| . . | . . |     ⇒
| . . | . . |     sends(l;e)
| . . | . . |     =
| . . | . . |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . | 19. ∀e:E.
| . . | . . |     loc(e) = i
| . . | . . |     ⇒
| . . | . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . | 20. ∀e:E.
| . . | . . |     isrcv(e)
| . . | . . |     ⇒
| . . | . . |     lnk(e) = l
| . . | . . |     ⇒
| . . | . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . | 21. e : E
| . . | . . | 22. loc(e) = i
| . . | . . | 23. kind(e) = k
| . . | . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . | 25. sends(l;e)
| . . | . . |     =
| . . | . . |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . | 27. L :
| . . | . . |     ∀n:ℕ||sends(l;e)||.
| . . | . . |     ∃e':E.
| . . | . . |     isrcv(e')
| . . | . . |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . | 28. LL :
| . . | . . |     n:ℕ||sends(l;e)||→{e':E
| . . | . . |                         | isrcv(e')
| . . | . . |                           & lnk(e') = l
| . . | . . |                             & sender(e') = e
| . . | . . |                             & index(e') = n }
| . . | . . | 29. (λn.1of(L(n))) = LL
| . . | . . | 30. val(e) ∈ ma-valtype(da; k)
| . . | . . | 31. (λz.(z when e)) ∈ State(ds)
| . . | . . | 32. {m:Msg((λl,tg. da(rcv(l; tg))?Top))
| . . | . . |      | mlnk(m) = l } List
| . . | . . | 33. (tg:Id
| . . | . . |      ×State(ds)→ma-valtype(da; k)→
| . . | . . |      (da(rcv(l; tg))?Void List)) List
| . . | . . | 34. tg:Id
| . . | . . |     ×State(ds)→ma-valtype(da; k)→
| . . | . . |     (da(rcv(l; tg))?Void List)
| . . | . . | 35. tg : Id
| . . | . . | 36. State(ds)
| . . | . . | 37. ma-valtype(da; k)
| . . | . . | 38. da(rcv(l; tg))?Void List
| . . | . . | 39. u1 : da(rcv(l; tg))?Void
| . . | . . | ⊢  u1 ∈ ma-valtype(da; rcv(l; tg))
| . . | . . |   by Unfold `ma-valtype` 0 THEN TrySubsume
| . . | . . |
```

```
| . . | . . .\
| . . | . . . .....subterm .....
| . . | . . . 1. i : Id
| . . | . . . 2. k : Knd
| . . | . . . 3. l : IdLnk
| . . | . . . 4. ds : x:Id fp-> Type
| . . | . . . 5. da : a:Knd fp-> Type
| . . | . . . 6. f :
| . . | . . .    (tg:Id
| . . | . . .     ×State(ds)→ma-valtype(da; k)→
| . . | . . .     (da(rcv(l; tg))?Void List)) List
| . . | . . . 7. source(l) = i
| . . | . . . 8. D' : Dsys
| . . | . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . 10. w : World
| . . | . . . 11. p : FairFifo
| . . | . . . 12. PossibleWorld(D';w)
| . . | . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . 14. ∀e:E.
| . . | . . .     loc(e) = i
| . . | . . .     ⇒
| . . | . . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . 15. ∀e:E.
| . . | . . .     isrcv(e)
| . . | . . .     ⇒
| . . | . . .     lnk(e) = l
| . . | . . .     ⇒
| . . | . . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . .     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . 17. ∀e:E.
| . . | . . .     loc(e) = i
| . . | . . .     ⇒
| . . | . . .     kind(e) = k
| . . | . . .     ⇒
| . . | . . .     sends(l;e)
| . . | . . .     =
| . . | . . .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . 19. ∀e:E.
| . . | . . .     loc(e) = i
| . . | . . .     ⇒
| . . | . . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . 20. ∀e:E.
| . . | . . .     isrcv(e)
| . . | . . .     ⇒
| . . | . . .     lnk(e) = l
| . . | . . .     ⇒
| . . | . . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . 21. e : E
| . . | . . . 22. loc(e) = i
| . . | . . . 23. kind(e) = k

| . . | . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . 25. sends(l;e)
| . . | . . .     =
```

```
| . . | . . .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . 27. L :
| . . | . . .     ∀n:ℕ||sends(l;e)||.
| . . | . . .     ∃e':E.
| . . | . . .     isrcv(e')
| . . | . . .     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . 28. LL :
| . . | . . .     n:ℕ||sends(l;e)||→{e':E
| . . | . . .                         | isrcv(e')
| . . | . . .                           & lnk(e') = l
| . . | . . .                             & sender(e') = e
| . . | . . .                             & index(e') = n }
| . . | . . . 29. (λn.1of(L(n))) = LL
| . . | . . . 30. val(e) ∈ ma-valtype(da; k)
| . . | . . . 31. (λz.(z when e)) ∈ State(ds)
| . . | . . . 32. {m:Msg((λl,tg. da(rcv(l; tg))?Top))
| . . | . . .      | mlnk(m) = l } List
| . . | . . . 33. m :
| . . | . . .     {m:Msg((λl,tg. da(rcv(l; tg))?Top))| mlnk(m) = l }
| . . | . . . ⊢  2of(m) ∈ tg:Id×ma-valtype(da; rcv(l; tg))
| . . | . . .   by Analyze -1 THEN Unfold `mlnk` -1 THEN Analyze -2
| . . | . . .      THEN
| . . | . . .      All Reduce
| . . | . . .      THEN
| . . | . . .      Unfold `ma-valtype` 0
| . . | . . .      THEN
| . . | . . .      DoSubsume
| . . | . . .      THEN
| . . | . . .      HypSubst' -1 0
| . . |
| . . |\
| . . | 1. i : Id
| . . | 2. k : Knd
| . . | 3. l : IdLnk
| . . | 4. ds : x:Id fp-> Type
| . . | 5. da : a:Knd fp-> Type
| . . | 6. f :
| . . | |  (tg:Id
| . . | |   ×State(ds)→ma-valtype(da; k)→
| . . | |   (da(rcv(l; tg))?Void List)) List
| . . | 7. source(l) = i
| . . | 8. D' : Dsys
| . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | 10. w : World
| . . | 11. p : FairFifo
| . . | 12. PossibleWorld(D';w)
| . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | 14. ∀e:E.
| . . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | 15. ∀e:E.
| . . | |   isrcv(e)
| . . | |   ⇒
| . . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
```

```
| . . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | 17. ∀e:E.
| . . | |   loc(e) = i
| . . | |   ⇒
| . . | |   kind(e) = k
| . . | |   ⇒
| . . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | 19. ∀e:E.
| . . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | 20. ∀e:E.
| . . | |   isrcv(e)
| . . | |   ⇒
| . . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | 21. e : E
| . . | 22. loc(e) = i
| . . | 23. kind(e) = k
| . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | 27. L :
| . . | |   ∀n:ℕ||sends(l;e)||.
| . . | |   ∃e':E.
| . . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | 28. LL :
| . . | |   n:ℕ||sends(l;e)||→{e':E
| . . | |                      | isrcv(e')
| . . | |                        & lnk(e') = l
| . . | |                          & sender(e') = e
| . . | |                          & index(e') = n }
| . . | 29. (λn.1of(L(n))) = LL
| . . | ⊢  map(λe'.<tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))
| . . | |  =
| . . | |  map(λm.2of(m);sends(l;e))
| . . | | by InstESAxiom ES(w) [] 9 THEN InstESAxiom ES(w) [] 16
| . . | |
| . . | 1. i : Id
| . . | 2. k : Knd
| . . | 3. l : IdLnk
| . . | 4. ds : x:Id fp-> Type
| . . | 5. da : a:Knd fp-> Type
| . . | 6. f :

| . . | |  (tg:Id
| . . | |   ×State(ds)→ma-valtype(da; k)→
| . . | |   (da(rcv(l; tg))?Void List)) List
| . . | 7. source(l) = i
| . . | 8. D' : Dsys
| . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | 10. w : World
| . . | 11. p : FairFifo
| . . | 12. PossibleWorld(D';w)
| . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | 14. ∀e:E.
| . . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | 15. ∀e:E.
```

```
| . . | |   isrcv(e)
| . . | |   ⇒
| . . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | 17. ∀e:E.
| . . | |   loc(e) = i
| . . | |   ⇒
| . . | |   kind(e) = k
| . . | |   ⇒
| . . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | 19. ∀e:E.
| . . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | 20. ∀e:E.
| . . | |   isrcv(e)
| . . | |   ⇒
| . . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | 21. e : E
| . . | 22. loc(e) = i
| . . | 23. kind(e) = k
| . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | 27. L :
| . . | |   ∀n:ℕ||sends(l;e)||.
| . . | |   ∃e':E.
| . . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | 28. LL :
| . . | |   n:ℕ||sends(l;e)||→{e':E
| . . | |                       | isrcv(e')
| . . | |                         & lnk(e') = l
| . . | |                           & sender(e') = e
| . . | |                           & index(e') = n }
| . . | 29. (λn.1of(L(n))) = LL
| . . | 30. ∀e:E.
| . . | |   isrcv(e)
| . . | |   ⇒
| . . | |   sends(lnk(e);sender(e))[index(e)]
| . . | |   =
| . . | |   msg(lnk(e);tag(e);val(e))
| . . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | |   ∃e':E.
| . . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | ⊢  map(λe'.<tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))
| . . | |  =
| . . | |  map(λm.2of(m);sends(l;e))
| . . | | by BackThru
| . . | |    Thm* ∀a,b:T List.
| . . | |         ||a|| = ||b|| ∈ ℤ
| . . | |         ⇒
| . . | |         (∀i:ℕ. i<||a|| ⇒ a[i] = b[i]) ⇒ a = b
| . . | |    {list_extensionality}
| . . | |    THEN
| . . | |    Try (Complete Auto)
```

```
| . . | |
| . . | |\
| . . | | .....wf .....
| . . | | 1. i : Id
| . . | | 2. k : Knd
| . . | | 3. l : IdLnk
| . . | | 4. ds : x:Id fp-> Type
| . . | | 5. da : a:Knd fp-> Type
| . . | | 6. f :
| . . | | |  (tg:Id
| . . | | |   ×State(ds)→ma-valtype(da; k)→
| . . | | |   (da(rcv(l; tg))?Void List)) List
| . . | | 7. source(l) = i
| . . | | 8. D' : Dsys
| . . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | 10. w : World
| . . | | 11. p : FairFifo
| . . | | 12. PossibleWorld(D';w)
| . . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | 14. ∀e:E.
| . . | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 15. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | 17. ∀e:E.
| . . | | |   loc(e) = i
| . . | | |   ⇒
| . . | | |   kind(e) = k
| . . | | |   ⇒
| . . | | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | 19. ∀e:E.
| . . | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 20. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 21. e : E
| . . | | 22. loc(e) = i
| . . | | 23. kind(e) = k
| . . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | 27. L :
| . . | | |   ∀n:ℕ||sends(l;e)||.
| . . | | |   ∃e':E.
| . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | 28. LL :
| . . | | |   n:ℕ||sends(l;e)||→{e':E
| . . | | |                       | isrcv(e')
| . . | | |                         & lnk(e') = l
| . . | | |                           & sender(e') = e
```

```
| . . | | |                              & index(e') = n }
| . . | | 29. (λn.1of(L(n))) = LL
| . . | | 30. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   sends(lnk(e);sender(e))[index(e)]
| . . | | |   =
| . . | | |   msg(lnk(e);tag(e);val(e))
| . . | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | |   ∃e':E.
| . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | ⊢  map(λe'.<tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))
| . . | | |  ∈ (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . | | | by GenConcl (map(LL;upto(||sends(l;e)||)) = X)
| . . | | |
| . . | | |\
| . . | | | .....wf .....
| . . | | | 1. i : Id
| . . | | | 2. k : Knd
| . . | | | 3. l : IdLnk
| . . | | | 4. ds : x:Id fp-> Type
| . . | | | 5. da : a:Knd fp-> Type
| . . | | | 6. f :
| . . | | | |  (tg:Id
| . . | | | |   ×State(ds)→ma-valtype(da; k)→
| . . | | | |   (da(rcv(l; tg))?Void List)) List
| . . | | | 7. source(l) = i
| . . | | | 8. D' : Dsys
| . . | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | | 10. w : World
| . . | | | 11. p : FairFifo
| . . | | | 12. PossibleWorld(D';w)
| . . | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 14. ∀e:E.
| . . | | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 15. ∀e:E.
| . . | | | |   isrcv(e)
| . . | | | |   ⇒
| . . | | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | | 17. ∀e:E.
| . . | | | |   loc(e) = i
| . . | | | |   ⇒
| . . | | | |   kind(e) = k
| . . | | | |   ⇒
| . . | | | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 19. ∀e:E.
| . . | | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 20. ∀e:E.
| . . | | | |   isrcv(e)
| . . | | | |   ⇒
| . . | | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 21. e : E
```

```
| . . | | | 22. loc(e) = i
| . . | | | 23. kind(e) = k
| . . | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 27. L :
| . . | | | |   ∀n:ℕ||sends(l;e)||.
| . . | | | |   ∃e':E.
| . . | | | |   isrcv(e')
| . . | | | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | 28. LL :
| . . | | | |   n:ℕ||sends(l;e)||→{e':E
| . . | | | |                       | isrcv(e')
| . . | | | |                         & lnk(e') = l
| . . | | | |                           & sender(e') = e
| . . | | | |                           & index(e') = n }
| . . | | | 29. (λn.1of(L(n))) = LL
| . . | | | 30. ∀e:E.
| . . | | | |   isrcv(e)
| . . | | | |   ⇒
| . . | | | |   sends(lnk(e);sender(e))[index(e)]
| . . | | | |   =
| . . | | | |   msg(lnk(e);tag(e);val(e))
| . . | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | | |   ∃e':E.
| . . | | | |   isrcv(e')
| . . | | | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | ⊢  map(LL;upto(||sends(l;e)||))
| . . | | | |  ∈ {e':E
| . . | | | |     | isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | | | by Auto
| . . | | | |
| . . | | | .....subterm .....
| . . | | | 1. i : Id
| . . | | | 2. k : Knd
| . . | | | 3. l : IdLnk
| . . | | | 4. ds : x:Id fp-> Type
| . . | | | 5. da : a:Knd fp-> Type
| . . | | | 6. f :
| . . | | |    (tg:Id
| . . | | |     ×State(ds)→ma-valtype(da; k)→
| . . | | |     (da(rcv(l; tg))?Void List)) List
| . . | | | 7. source(l) = i
| . . | | | 8. D' : Dsys
| . . | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | | 10. w : World
| . . | | | 11. p : FairFifo
| . . | | | 12. PossibleWorld(D';w)
| . . | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 14. ∀e:E.
| . . | | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 15. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
```

```
| . . | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | | | 17. ∀e:E.
| . . | | |     loc(e) = i
| . . | | |     ⇒
| . . | | |     kind(e) = k
| . . | | |     ⇒
| . . | | |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 19. ∀e:E.
| . . | | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 20. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 21. e : E
| . . | | | 22. loc(e) = i
| . . | | | 23. kind(e) = k
| . . | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 27. L :
| . . | | |     ∀n:ℕ||sends(l;e)||.
| . . | | |     ∃e':E.
| . . | | |     isrcv(e')
| . . | | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | 28. LL :
| . . | | |     n:ℕ||sends(l;e)||→{e':E
| . . | | |                           | isrcv(e')
| . . | | |                             & lnk(e') = l
| . . | | |                               & sender(e') = e
| . . | | |                               & index(e') = n }
| . . | | | 29. (λn.1of(L(n))) = LL
| . . | | | 30. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     sends(lnk(e);sender(e))[index(e)]
| . . | | |     =
| . . | | |     msg(lnk(e);tag(e);val(e))
| . . | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | |     ∃e':E.
| . . | | |     isrcv(e')
| . . | | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | ⊢  LL
| . . | | |    ∈ ℕ||sends(l;e)||→{e':E
| . . | | |                          | isrcv(e')
| . . | | |                            & lnk(e') = l & sender(e') = e }
| . . | | |   by Ext2 THEN DoSubsume THEN Analyze 0 THEN Analyze -1
| . . | | |      THEN
| . . | | |      Analyze
| . . | | |      THEN
| . . | | |      ExRepD
| . . | | |
| . . | | |\
| . . | | | .....wf .....
```

```
| . . | | | 1. i : Id
| . . | | | 2. k : Knd
| . . | | | 3. l : IdLnk
| . . | | | 4. ds : x:Id fp-> Type
| . . | | | 5. da : a:Knd fp-> Type
| . . | | | 6. f :
| . . | | |    (tg:Id
| . . | | |     ×State(ds)→ma-valtype(da; k)→
| . . | | |     (da(rcv(l; tg))?Void List)) List
| . . | | | 7. source(l) = i
| . . | | | 8. D' : Dsys
| . . | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | | 10. w : World
| . . | | | 11. p : FairFifo
| . . | | | 12. PossibleWorld(D';w)
| . . | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 14. ∀e:E.
| . . | | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 15. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | | | 17. ∀e:E.
| . . | | |     loc(e) = i
| . . | | |     ⇒
| . . | | |     kind(e) = k
| . . | | |     ⇒
| . . | | |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 19. ∀e:E.
| . . | | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 20. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 21. e : E
| . . | | | 22. loc(e) = i
| . . | | | 23. kind(e) = k
| . . | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 27. L :
| . . | | |     ∀n:ℕ||sends(l;e)||.
| . . | | |     ∃e':E.
| . . | | |     isrcv(e')
| . . | | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | 28. LL :
| . . | | |     n:ℕ||sends(l;e)||→{e':E
| . . | | |                          | isrcv(e')
| . . | | |                            & lnk(e') = l
| . . | | |                              & sender(e') = e
| . . | | |                              & index(e') = n }
| . . | | | 29. (λn.1of(L(n))) = LL
```

```
| . . | | | 30. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     sends(lnk(e);sender(e))[index(e)]
| . . | | |     =
| . . | | |     msg(lnk(e);tag(e);val(e))
| . . | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | |     ∃e':E.
| . . | | |     isrcv(e')
| . . | | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | ⊢  {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | | |    ∈ Type  by Auto
| . . | | |
| . . | | .\
| . . | | . 1. i : Id
| . . | | . 2. k : Knd
| . . | | . 3. l : IdLnk
| . . | | . 4. ds : x:Id fp-> Type
| . . | | . 5. da : a:Knd fp-> Type
| . . | | . 6. f :
| . . | | . |  (tg:Id
| . . | | . |   ×State(ds)→ma-valtype(da; k)→
| . . | | . |   (da(rcv(l; tg))?Void List)) List
| . . | | . 7. source(l) = i
| . . | | . 8. D' : Dsys
| . . | | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | . 10. w : World
| . . | | . 11. p : FairFifo
| . . | | . 12. PossibleWorld(D';w)
| . . | | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 14. ∀e:E.
| . . | | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 15. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | . 17. ∀e:E.
| . . | | . |   loc(e) = i
| . . | | . |   ⇒
| . . | | . |   kind(e) = k
| . . | | . |   ⇒
| . . | | . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 19. ∀e:E.
| . . | | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 20. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 21. e : E
| . . | | . 22. loc(e) = i
| . . | | . 23. kind(e) = k
| . . | | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
```

```
| . . | | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 27. L :
| . . | | . |   ∀n:ℕ||sends(l;e)||.
| . . | | . |   ∃e':E.
| . . | | . |   isrcv(e')
| . . | | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 28. LL :
| . . | | . |   n:ℕ||sends(l;e)||→{e':E
| . . | | . |                     | isrcv(e')
| . . | | . |                       & lnk(e') = l
| . . | | . |                         & sender(e') = e
| . . | | . |                         & index(e') = n }
| . . | | . 29. (λn.1of(L(n))) = LL
| . . | | . 30. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   sends(lnk(e);sender(e))[index(e)]
| . . | | . |   =
| . . | | . |   msg(lnk(e);tag(e);val(e))
| . . | | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | . |   ∃e':E.
| . . | | . |   isrcv(e')
| . . | | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 32. X :
| . . | | . |   {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | | . 33. map(LL;upto(||sends(l;e)||)) = X
| . . | | . ⊢  map(λe'.<tag(e'),val(e')>;X)
| . . | | . |  ∈ (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . | | . | by Auto THEN Analyze -1 THEN ExRepD THEN Unhide
| . . | | . |    THEN
| . . | | . |    ExRepD
| . . | | . |    THEN
| . . | | . |    DoSubsume
| . . | | . |
| . . | | . .....subtype_rel .....
| . . | | . 1. i : Id
| . . | | . 2. k : Knd
| . . | | . 3. l : IdLnk
| . . | | . 4. ds : x:Id fp-> Type
| . . | | . 5. da : a:Knd fp-> Type
| . . | | . 6. f :
| . . | | . |  (tg:Id
| . . | | . |   ×State(ds)→ma-valtype(da; k)→
| . . | | . |   (da(rcv(l; tg))?Void List)) List
| . . | | . 7. source(l) = i
| . . | | . 8. D' : Dsys
| . . | | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | . 10. w : World
| . . | | . 11. p : FairFifo
| . . | | . 12. PossibleWorld(D';w)
| . . | | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 14. ∀e:E.
| . . | | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 15. ∀e:E.
```

```
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | . 17. ∀e:E.
| . . | | . |   loc(e) = i
| . . | | . |   ⇒
| . . | | . |   kind(e) = k
| . . | | . |   ⇒
| . . | | . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 19. ∀e:E.
| . . | | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 20. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 21. e : E
| . . | | . 22. loc(e) = i
| . . | | . 23. kind(e) = k
| . . | | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 27. L :
| . . | | . |   ∀n:ℕ||sends(l;e)||.
| . . | | . |   ∃e':E.
| . . | | . |   isrcv(e')
| . . | | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 28. LL :
| . . | | . |   n:ℕ||sends(l;e)||→{e':E
| . . | | . |                      | isrcv(e')
| . . | | . |                        & lnk(e') = l
| . . | | . |                          & sender(e') = e
| . . | | . |                          & index(e') = n }
| . . | | . 29. (λn.1of(L(n))) = LL
| . . | | . 30. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   sends(lnk(e);sender(e))[index(e)]
| . . | | . |   =
| . . | | . |   msg(lnk(e);tag(e);val(e))
| . . | | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | . |   ∃e':E.
| . . | | . |   isrcv(e')
| . . | | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 32. X :
| . . | | . |   {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | | . 33. map(LL;upto(||sends(l;e)||)) = X
| . . | | . 34. e' : E
| . . | | . 35. isrcv(e')
| . . | | . 36. lnk(e') = l
| . . | | . 37. sender(e') = e
| . . | | . ⊢  valtype(e') ⊆r ma-valtype(da; rcv(l; tag(e')))
| . . | | . | by Subst' (rcv(l; tag(e')) = kind(e')) 0
```

```
| . . | | . |
| . . | | . |\
| . . | | . | 1. i : Id
| . . | | . | 2. k : Knd
| . . | | . | 3. l : IdLnk
| . . | | . | 4. ds : x:Id fp-> Type
| . . | | . | 5. da : a:Knd fp-> Type
| . . | | . | 6. f :
| . . | | . |    (tg:Id
| . . | | . |     ×State(ds)→ma-valtype(da; k)→
| . . | | . |     (da(rcv(l; tg))?Void List)) List
| . . | | . | 7. source(l) = i
| . . | | . | 8. D' : Dsys
| . . | | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | . | 10. w : World
| . . | | . | 11. p : FairFifo
| . . | | . | 12. PossibleWorld(D';w)
| . . | | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . | 14. ∀e:E.
| . . | | . |     loc(e) = i
| . . | | . |     ⇒
| . . | | . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . | 15. ∀e:E.
| . . | | . |     isrcv(e)
| . . | | . |     ⇒
| . . | | . |     lnk(e) = l
| . . | | . |     ⇒
| . . | | . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | . |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | | . | 17. ∀e:E.
| . . | | . |     loc(e) = i
| . . | | . |     ⇒
| . . | | . |     kind(e) = k
| . . | | . |     ⇒
| . . | | . |     sends(l;e)
| . . | | . |     =
| . . | | . |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . | 19. ∀e:E.
| . . | | . |     loc(e) = i
| . . | | . |     ⇒
| . . | | . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . | 20. ∀e:E.
| . . | | . |     isrcv(e)
| . . | | . |     ⇒
| . . | | . |     lnk(e) = l
| . . | | . |     ⇒
| . . | | . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . | 21. e : E
| . . | | . | 22. loc(e) = i
| . . | | . | 23. kind(e) = k
| . . | | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . | 25. sends(l;e)
| . . | | . |     =
```

```
| . . | | . |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . | 27. L :
| . . | | . |     ∀n:ℕ||sends(l;e)||.
| . . | | . |     ∃e':E.
| . . | | . |     isrcv(e')
| . . | | . |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . | 28. LL :
| . . | | . |     n:ℕ||sends(l;e)||→{e':E
| . . | | . |                         | isrcv(e')
| . . | | . |                           & lnk(e') = l
| . . | | . |                             & sender(e') = e
| . . | | . |                             & index(e') = n }
| . . | | . | 29. (λn.1of(L(n))) = LL
| . . | | . | 30. ∀e:E.
| . . | | . |     isrcv(e)
| . . | | . |     ⇒
| . . | | . |     sends(lnk(e);sender(e))[index(e)]
| . . | | . |     =
| . . | | . |     msg(lnk(e);tag(e);val(e))
| . . | | . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | . |     ∃e':E.
| . . | | . |     isrcv(e')
| . . | | . |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . | 32. X :
| . . | | . |     {e':E
| . . | | . |      | isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | | . | 33. map(LL;upto(||sends(l;e)||)) = X
| . . | | . | 34. e' : E
| . . | | . | 35. isrcv(e')
| . . | | . | 36. lnk(e') = l
| . . | | . | 37. sender(e') = e
| . . | | . | ⊢  rcv(l; tag(e')) = kind(e')
| . . | | . |   by MoveToConcl -2 THEN MoveToConcl -2
| . . | | . |      THEN
| . . | | . |      Unfolds [`es-isrcv`;`es-lnk`;`es-tag`] 0
| . . | | . |      THEN

| . . | | . |      GenConclAtAddr [2;2;3]
| . . | | . |      THEN
| . . | | . |      Analyze -2
| . . | | . |      THEN
| . . | | . |      Analyze -2
| . . | | . |      THEN
| . . | | . |      Repeat
| . . | | . |      (Unfolds [`isrcv`;`lnk`;`tagof`] 0 THEN Reduce 0)
| . . | | . |      THEN
| . . | | . |      Fold `rcv` 0
| . . | | . |      THEN
| . . | | . |      Analyze
| . . | | . |
| . . | | . .\
| . . | | . . 1. i : Id
| . . | | . . 2. k : Knd
| . . | | . . 3. l : IdLnk
| . . | | . . 4. ds : x:Id fp-> Type
```

```
| . . | | . . 5. da : a:Knd fp-> Type
| . . | | . . 6. f :
| . . | | . .    (tg:Id
| . . | | . .     ×State(ds)→ma-valtype(da; k)→
| . . | | . .     (da(rcv(l; tg))?Void List)) List
| . . | | . . 7. source(l) = i
| . . | | . . 8. D' : Dsys
| . . | | . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | . . 10. w : World
| . . | | . . 11. p : FairFifo
| . . | | . . 12. PossibleWorld(D';w)
| . . | | . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . . 14. ∀e:E.
| . . | | . .     loc(e) = i
| . . | | . .     ⇒
| . . | | . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . . 15. ∀e:E.
| . . | | . .     isrcv(e)
| . . | | . .     ⇒
| . . | | . .     lnk(e) = l
| . . | | . .     ⇒
| . . | | . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | . .     ((λl,tg. da(rcv(l; tg))?Top))
| . . | | . . 17. ∀e:E.
| . . | | . .     loc(e) = i
| . . | | . .     ⇒
| . . | | . .     kind(e) = k
| . . | | . .     ⇒
| . . | | . .     sends(l;e)
| . . | | . .     =
| . . | | . .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . . 19. ∀e:E.
| . . | | . .     loc(e) = i
| . . | | . .     ⇒
| . . | | . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . . 20. ∀e:E.
| . . | | . .     isrcv(e)
| . . | | . .     ⇒
| . . | | . .     lnk(e) = l
| . . | | . .     ⇒
| . . | | . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . . 21. e : E
| . . | | . . 22. loc(e) = i
| . . | | . . 23. kind(e) = k
| . . | | . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . . 25. sends(l;e)
| . . | | . .     =
| . . | | . .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . . 27. L :
| . . | | . .     ∀n:ℕ||sends(l;e)||.
| . . | | . .     ∃e':E.
| . . | | . .     isrcv(e')
```

```
| . . | | . .     & lnk(e’) = l & sender(e’) = e & index(e’) = n
| . . | | . . 28. LL :
| . . | | . .     n:ℕ||sends(l;e)||→{e’:E
| . . | | . .                         | isrcv(e’)
| . . | | . .                           & lnk(e’) = l
| . . | | . .                             & sender(e’) = e
| . . | | . .                             & index(e’) = n }
| . . | | . . 29. (λn.1of(L(n))) = LL
| . . | | . . 30. ∀e:E.
| . . | | . .     isrcv(e)
| . . | | . .     ⇒
| . . | | . .     sends(lnk(e);sender(e))[index(e)]
| . . | | . .     =
| . . | | . .     msg(lnk(e);tag(e);val(e))
| . . | | . . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | . .     ∃e’:E.
| . . | | . .     isrcv(e’)
| . . | | . .     & lnk(e’) = l & sender(e’) = e & index(e’) = n
| . . | | . . 32. X :
| . . | | . .     {e’:E
| . . | | . .      | isrcv(e’) & lnk(e’) = l & sender(e’) = e } List
| . . | | . . 33. map(LL;upto(||sends(l;e)||)) = X
| . . | | . . 34. e’ : E
| . . | | . . 35. isrcv(e’)
| . . | | . . 36. lnk(e’) = l
| . . | | . . 37. sender(e’) = e
| . . | | . . ⊢  valtype(e’) ⊆r ma-valtype(da; kind(e’))

| . . | | . .   by BackThruSomeHyp
| . . | |
| . . | |\
| . . | | .....wf .....
| . . | | 1. i : Id
| . . | | 2. k : Knd
| . . | | 3. l : IdLnk
| . . | | 4. ds : x:Id fp-> Type
| . . | | 5. da : a:Knd fp-> Type
| . . | | 6. f :
| . . | | |  (tg:Id
| . . | | |   ×State(ds)→ma-valtype(da; k)→
| . . | | |   (da(rcv(l; tg))?Void List)) List
| . . | | 7. source(l) = i
| . . | | 8. D’ : Dsys
| . . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D’
| . . | | 10. w : World
| . . | | 11. p : FairFifo
| . . | | 12. PossibleWorld(D’;w)
| . . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | 14. ∀e:E.
| . . | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 15. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | |   ((λl,tg. da(rcv(l; tg))?Top))
```

```
| . . | | 17. ∀e:E.
| . . | | |   loc(e) = i
| . . | | |   ⇒
| . . | | |   kind(e) = k
| . . | | |   ⇒
| . . | | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | 19. ∀e:E.
| . . | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 20. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 21. e : E
| . . | | 22. loc(e) = i
| . . | | 23. kind(e) = k
| . . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | 27. L :
| . . | | |   ∀n:ℕ||sends(l;e)||.
| . . | | |   ∃e':E.
| . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | 28. LL :
| . . | | |   n:ℕ||sends(l;e)||→{e':E
| . . | | |                     | isrcv(e')
| . . | | |                       & lnk(e') = l
| . . | | |                         & sender(e') = e
| . . | | |                         & index(e') = n }
| . . | | 29. (λn.1of(L(n))) = LL
| . . | | 30. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   sends(lnk(e);sender(e))[index(e)]
| . . | | |   =
| . . | | |   msg(lnk(e);tag(e);val(e))
| . . | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | |   ∃e':E.
| . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | ⊢  map(λm.2of(m);sends(l;e))
| . . | | |  ∈ (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . | | | by Auto THEN Analyze -1 THEN Unfold `haslink` -1
| . . | | |
| . . | | 1. i : Id
| . . | | 2. k : Knd
| . . | | 3. l : IdLnk
| . . | | 4. ds : x:Id fp-> Type
| . . | | 5. da : a:Knd fp-> Type
| . . | | 6. f :
| . . | | |  (tg:Id
| . . | | |   ×State(ds)→ma-valtype(da; k)→
| . . | | |   (da(rcv(l; tg))?Void List)) List
| . . | | 7. source(l) = i
| . . | | 8. D' : Dsys
| . . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
```

```
| . . | | 10. w : World
| . . | | 11. p : FairFifo
| . . | | 12. PossibleWorld(D';w)
| . . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | 14. ∀e:E.
| . . | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 15. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | 17. ∀e:E.
| . . | | |   loc(e) = i
| . . | | |   ⇒
| . . | | |   kind(e) = k
| . . | | |   ⇒
| . . | | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | 19. ∀e:E.
| . . | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 20. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 21. e : E
| . . | | 22. loc(e) = i
| . . | | 23. kind(e) = k
| . . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | 27. L :
| . . | | |   ∀n:ℕ||sends(l;e)||.
| . . | | |   ∃e':E.
| . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | 28. LL :
| . . | | |   n:ℕ||sends(l;e)||→{e':E
| . . | | |                       | isrcv(e')
| . . | | |                         & lnk(e') = l
| . . | | |                           & sender(e') = e
| . . | | |                           & index(e') = n }
| . . | | 29. (λn.1of(L(n))) = LL
| . . | | 30. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   sends(lnk(e);sender(e))[index(e)]
| . . | | |   =
| . . | | |   msg(lnk(e);tag(e);val(e))
| . . | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | |   ∃e':E.
| . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | 32. m : Msg
| . . | | 33. mlnk(m) = l
| . . | ⊢  2of(m) ∈ tg:Id×ma-valtype(da; rcv(l; tg))
| . . | | | by Assert (m ∈ Msg((λl,tg. da(rcv(l; tg))?Top)))
```

```
| . . | | |
| . . | | |\
| . . | | | .....assertion .....
| . . | | | 1. i : Id
| . . | | | 2. k : Knd
| . . | | | 3. l : IdLnk
| . . | | | 4. ds : x:Id fp-> Type
| . . | | | 5. da : a:Knd fp-> Type
| . . | | | 6. f :
| . . | | | |  (tg:Id
| . . | | | |   ×State(ds)→ma-valtype(da; k)→
| . . | | | |   (da(rcv(l; tg))?Void List)) List
| . . | | | 7. source(l) = i
| . . | | | 8. D' : Dsys
| . . | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | | 10. w : World
| . . | | | 11. p : FairFifo
| . . | | | 12. PossibleWorld(D';w)
| . . | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 14. ∀e:E.
| . . | | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 15. ∀e:E.
| . . | | | |   isrcv(e)
| . . | | | |   ⇒
| . . | | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | | 17. ∀e:E.
| . . | | | |   loc(e) = i
| . . | | | |   ⇒
| . . | | | |   kind(e) = k
| . . | | | |   ⇒
| . . | | | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 19. ∀e:E.
| . . | | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 20. ∀e:E.
| . . | | | |   isrcv(e)
| . . | | | |   ⇒
| . . | | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 21. e : E
| . . | | | 22. loc(e) = i
| . . | | | 23. kind(e) = k
| . . | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 27. L :
| . . | | | |   ∀n:ℕ||sends(l;e)||.
| . . | | | |   ∃e':E.
| . . | | | |   isrcv(e')
| . . | | | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | 28. LL :
| . . | | | |   n:ℕ||sends(l;e)||→{e':E
| . . | | | |                       | isrcv(e')
| . . | | | |                         & lnk(e') = l
```

```
| . . | | | |                           & sender(e') = e
| . . | | | |                           & index(e') = n }
| . . | | | 29. (λn.1of(L(n))) = LL
| . . | | | 30. ∀e:E.
| . . | | | |   isrcv(e)
| . . | | | |   ⇒
| . . | | | |   sends(lnk(e);sender(e))[index(e)]
| . . | | | |   =
| . . | | | |   msg(lnk(e);tag(e);val(e))
| . . | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | | |   ∃e':E.
| . . | | | |   isrcv(e')
| . . | | | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | 32. m : Msg
| . . | | | 33. mlnk(m) = l
| . . | | | ⊢  m ∈ Msg((λl,tg. da(rcv(l; tg))?Top))
| . . | | | | by Assert (m ∈ {m:Msg| source(mlnk(m)) = i })
| . . | | | |    THEN
| . . | | | |    TrySubsume
| . . | | | |
| . . | | | .....assertion .....
| . . | | | 1. i : Id
| . . | | | 2. k : Knd
| . . | | | 3. l : IdLnk
| . . | | | 4. ds : x:Id fp-> Type
| . . | | | 5. da : a:Knd fp-> Type
| . . | | | 6. f :
| . . | | | |  (tg:Id
| . . | | | |   ×State(ds)→ma-valtype(da; k)→
| . . | | | |   (da(rcv(l; tg))?Void List)) List
| . . | | | 7. source(l) = i
| . . | | | 8. D' : Dsys
| . . | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | | 10. w : World
| . . | | | 11. p : FairFifo
| . . | | | 12. PossibleWorld(D';w)
| . . | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 14. ∀e:E.
| . . | | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 15. ∀e:E.
| . . | | | |   isrcv(e)
| . . | | | |   ⇒
| . . | | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | | 17. ∀e:E.
| . . | | | |   loc(e) = i
| . . | | | |   ⇒
| . . | | | |   kind(e) = k
| . . | | | |   ⇒
| . . | | | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 19. ∀e:E.
| . . | | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 20. ∀e:E.
```

```
| . . | | | |   isrcv(e)
| . . | | | |   ⇒
| . . | | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 21. e : E
| . . | | | 22. loc(e) = i
| . . | | | 23. kind(e) = k
| . . | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 27. L :
| . . | | | |   ∀n:ℕ||sends(l;e)||.
| . . | | | |   ∃e':E.
| . . | | | |   isrcv(e')
| . . | | | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | 28. LL :
| . . | | | |   n:ℕ||sends(l;e)||→{e':E
| . . | | | |                        | isrcv(e')
| . . | | | |                          & lnk(e') = l
| . . | | | |                            & sender(e') = e
| . . | | | |                            & index(e') = n }
| . . | | | 29. (λn.1of(L(n))) = LL
| . . | | | 30. ∀e:E.
| . . | | | |   isrcv(e)
| . . | | | |   ⇒
| . . | | | |   sends(lnk(e);sender(e))[index(e)]
| . . | | | |   =
| . . | | | |   msg(lnk(e);tag(e);val(e))
| . . | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | | |   ∃e':E.
| . . | | | |   isrcv(e')
| . . | | | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | 32. m : Msg
| . . | | | 33. mlnk(m) = l
| . . | | | ⊢  m ∈ {m:Msg| source(mlnk(m)) = i }  by Analyze
| . . | | | |
| . . | | | .....set predicate .....
| . . | | | 1. i : Id
| . . | | | 2. k : Knd
| . . | | | 3. l : IdLnk
| . . | | | 4. ds : x:Id fp-> Type
| . . | | | 5. da : a:Knd fp-> Type
| . . | | | 6. f :
| . . | | |    (tg:Id
| . . | | |     ×State(ds)→ma-valtype(da; k)→
| . . | | |     (da(rcv(l; tg))?Void List)) List
| . . | | | 7. source(l) = i
| . . | | | 8. D' : Dsys
| . . | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | | 10. w : World
| . . | | | 11. p : FairFifo
| . . | | | 12. PossibleWorld(D';w)
| . . | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top

| . . | | | 14. ∀e:E.
| . . | | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 15. ∀e:E.
```

```
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | | | 17. ∀e:E.
| . . | | |     loc(e) = i
| . . | | |     ⇒
| . . | | |     kind(e) = k
| . . | | |     ⇒
| . . | | |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 19. ∀e:E.
| . . | | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 20. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 21. e : E
| . . | | | 22. loc(e) = i
| . . | | | 23. kind(e) = k
| . . | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 27. L :
| . . | | |     ∀n:ℕ||sends(l;e)||.
| . . | | |     ∃e':E.
| . . | | |     isrcv(e')
| . . | | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | 28. LL :
| . . | | |     n:ℕ||sends(l;e)||→{e':E
| . . | | |                         | isrcv(e')
| . . | | |                           & lnk(e') = l
| . . | | |                             & sender(e') = e
| . . | | |                             & index(e') = n }
| . . | | | 29. (λn.1of(L(n))) = LL
| . . | | | 30. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     sends(lnk(e);sender(e))[index(e)]
| . . | | |     =
| . . | | |     msg(lnk(e);tag(e);val(e))
| . . | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | |     ∃e':E.
| . . | | |     isrcv(e')
| . . | | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | 32. m : Msg
| . . | | | 33. mlnk(m) = l
| . . | | | ⊢  source(mlnk(m)) = i  by HypSubst' -1 0
| . . | | |
| . . | | .\
| . . | | . 1. i : Id
| . . | | . 2. k : Knd
| . . | | . 3. l : IdLnk
| . . | | . 4. ds : x:Id fp-> Type
```

```
| . . | | . 5. da : a:Knd fp-> Type
| . . | | . 6. f :
| . . | | . |  (tg:Id
| . . | | . |   ×State(ds)→ma-valtype(da; k)→
| . . | | . |   (da(rcv(l; tg))?Void List)) List
| . . | | . 7. source(l) = i
| . . | | . 8. D' : Dsys
| . . | | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | . 10. w : World
| . . | | . 11. p : FairFifo
| . . | | . 12. PossibleWorld(D';w)
| . . | | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 14. ∀e:E.
| . . | | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 15. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | . 17. ∀e:E.
| . . | | . |   loc(e) = i
| . . | | . |   ⇒
| . . | | . |   kind(e) = k
| . . | | . |   ⇒
| . . | | . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 19. ∀e:E.
| . . | | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 20. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 21. e : E
| . . | | . 22. loc(e) = i
| . . | | . 23. kind(e) = k
| . . | | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 27. L :
| . . | | . |   ∀n:ℕ||sends(l;e)||.
| . . | | . |   ∃e':E.

| . . | | . |   isrcv(e')
| . . | | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 28. LL :
| . . | | . |   n:ℕ||sends(l;e)||→{e':E
| . . | | . |                       | isrcv(e')
| . . | | . |                         & lnk(e') = l
| . . | | . |                           & sender(e') = e
| . . | | . |                           & index(e') = n }
| . . | | . 29. (λn.1of(L(n))) = LL
| . . | | . 30. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   sends(lnk(e);sender(e))[index(e)]
```

```
| . . | | . |    =
| . . | | . |    msg(lnk(e);tag(e);val(e))
| . . | | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | . |    ∃e':E.
| . . | | . |    isrcv(e')
| . . | | . |    & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 32. m : Msg
| . . | | . 33. mlnk(m) = l
| . . | | . 34. m ∈ Msg((λl,tg. da(rcv(l; tg))?Top))
| . . | | . ⊢  2of(m) ∈ tg:Id×ma-valtype(da; rcv(l; tg))
| . . | | . | by GenConclAtAddr [2;1] THEN Analyze -2 THEN Analyze -2
| . . | | . |     THEN
| . . | | . |     All Reduce
| . . | | . |     THEN
| . . | | . |     Unfold `ma-valtype` 0
| . . | | . |     THEN
| . . | | . |     DoSubsume
| . . | | . |
| . . | | . .....subtype_rel .....
| . . | | . 1. i : Id
| . . | | . 2. k : Knd
| . . | | . 3. l : IdLnk
| . . | | . 4. ds : x:Id fp-> Type
| . . | | . 5. da : a:Knd fp-> Type
| . . | | . 6. f :
| . . | | .    (tg:Id
| . . | | .     ×State(ds)→ma-valtype(da; k)→
| . . | | .     (da(rcv(l; tg))?Void List)) List
| . . | | . 7. source(l) = i
| . . | | . 8. D' : Dsys
| . . | | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | . 10. w : World
| . . | | . 11. p : FairFifo
| . . | | . 12. PossibleWorld(D';w)
| . . | | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 14. ∀e:E.
| . . | | .     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 15. ∀e:E.
| . . | | .     isrcv(e)
| . . | | .     ⇒
| . . | | .     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | .     ((λl,tg. da(rcv(l; tg))?Top))
| . . | | . 17. ∀e:E.
| . . | | .     loc(e) = i
| . . | | .     ⇒
| . . | | .     kind(e) = k
| . . | | .     ⇒
| . . | | .     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 19. ∀e:E.
| . . | | .     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 20. ∀e:E.
| . . | | .     isrcv(e)
| . . | | .     ⇒
```

```
| . . | | .     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 21. e : E
| . . | | . 22. loc(e) = i
| . . | | . 23. kind(e) = k
| . . | | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 27. L :
| . . | | .     ∀n:ℕ||sends(l;e)||.
| . . | | .     ∃e':E.
| . . | | .     isrcv(e')
| . . | | .     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 28. LL :
| . . | | .     n:ℕ||sends(l;e)||→{e':E
| . . | | .                         | isrcv(e')
| . . | | .                           & lnk(e') = l
| . . | | .                             & sender(e') = e
| . . | | .                             & index(e') = n }
| . . | | . 29. (λn.1of(L(n))) = LL
| . . | | . 30. ∀e:E.
| . . | | .     isrcv(e)
| . . | | .     ⇒
| . . | | .     sends(lnk(e);sender(e))[index(e)]
| . . | | .     =
| . . | | .     msg(lnk(e);tag(e);val(e))
| . . | | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | .     ∃e':E.
| . . | | .     isrcv(e')
| . . | | .     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 32. m : Msg
| . . | | . 33. mlnk(m) = l
| . . | | . 34. m ∈ Msg((λl,tg. da(rcv(l; tg))?Top))
| . . | | . 35. l1 : IdLnk
| . . | | . 36. t : Id
| . . | | . 37. v2 : da(rcv(l1; t))?Top
| . . | | . 38. m = <l1,t,v2>
| . . | | . ⊢  da(rcv(l1; t))?Top ⊆r da(rcv(l; t))?Top
| . . | | .   by MoveToHyp -1 -6 THEN HypSubst -1 -2
| . . | | .      THEN
| . . | | .      Unfold `mlnk` -2
| . . | | .      THEN
| . . | | .      Reduce -2
| . . | | .      THEN
| . . | | .      HypSubst' -2 0
| . . | |
| . . | |\
| . . | | 1. i : Id
| . . | | 2. k : Knd
| . . | | 3. l : IdLnk
| . . | | 4. ds : x:Id fp-> Type
| . . | | 5. da : a:Knd fp-> Type
| . . | | 6. f :
| . . | | |  (tg:Id
| . . | | |   ×State(ds)→ma-valtype(da; k)→
| . . | | |   (da(rcv(l; tg))?Void List)) List
```

```
| . . | | 7. source(l) = i
| . . | | 8. D' : Dsys
| . . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | 10. w : World
| . . | | 11. p : FairFifo
| . . | | 12. PossibleWorld(D';w)
| . . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | 14. ∀e:E.
| . . | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 15. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | 17. ∀e:E.
| . . | | |   loc(e) = i
| . . | | |   ⇒
| . . | | |   kind(e) = k
| . . | | |   ⇒
| . . | | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | 19. ∀e:E.
| . . | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 20. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 21. e : E
| . . | | 22. loc(e) = i
| . . | | 23. kind(e) = k
| . . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | 27. L :
| . . | | |   ∀n:ℕ||sends(l;e)||.
| . . | | |   ∃e':E.
| . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | 28. LL :
| . . | | |   n:ℕ||sends(l;e)||→{e':E
| . . | | |                     | isrcv(e')
| . . | | |                       & lnk(e') = l
| . . | | |                         & sender(e') = e
| . . | | |                         & index(e') = n }
| . . | | 29. (λn.1of(L(n))) = LL
| . . | | 30. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   sends(lnk(e);sender(e))[index(e)]
| . . | | |   =
| . . | | |   msg(lnk(e);tag(e);val(e))
| . . | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | |   ∃e':E.
| . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | ⊢  ||map(λe'.
```

```
| . . | | |  <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | | |  =
| . . | | |  ||map(λm.2of(m);sends(l;e))||
| . . | | | by RWO
| . . | | |    Thm* ∀f:Top, T:Type, L:T List. ||map(f;L)|| ~ ||L||
| . . | | |    {length_DASH_map}
| . . | | |    0
| . . | | |
| . . | | 1. i : Id
| . . | | 2. k : Knd
| . . | | 3. l : IdLnk
| . . | | 4. ds : x:Id fp-> Type
| . . | | 5. da : a:Knd fp-> Type
| . . | | 6. f :
| . . | | |  (tg:Id
| . . | | |   ×State(ds)→ma-valtype(da; k)→
| . . | | |   (da(rcv(l; tg))?Void List)) List
| . . | | 7. source(l) = i
| . . | | 8. D' : Dsys
| . . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | 10. w : World
| . . | | 11. p : FairFifo
| . . | | 12. PossibleWorld(D';w)
| . . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | 14. ∀e:E.
| . . | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 15. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | 17. ∀e:E.
| . . | | |   loc(e) = i
| . . | | |   ⇒
| . . | | |   kind(e) = k
| . . | | |   ⇒
| . . | | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | 19. ∀e:E.
| . . | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 20. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | 21. e : E
| . . | | 22. loc(e) = i
| . . | | 23. kind(e) = k
| . . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | 27. L :
| . . | | |   ∀n:ℕ||sends(l;e)||.
| . . | | |   ∃e':E.
| . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
```

```
| . . | | 28. LL :
| . . | | |   n:ℕ||sends(l;e)||→{e':E
| . . | | |                         | isrcv(e')
| . . | | |                           & lnk(e') = l
| . . | | |                             & sender(e') = e
| . . | | |                             & index(e') = n }
| . . | | 29. (λn.1of(L(n))) = LL
| . . | | 30. ∀e:E.
| . . | | |   isrcv(e)
| . . | | |   ⇒
| . . | | |   sends(lnk(e);sender(e))[index(e)]
| . . | | |   =
| . . | | |   msg(lnk(e);tag(e);val(e))
| . . | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | |   ∃e':E.
| . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | ⊢  ||map(λe'.
| . . | | |  <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | | |  =
| . . | | |  ||sends(l;e)||
| . . | | | by GenConcl (map(LL;upto(||sends(l;e)||)) = X)
| . . | | |
| . . | | |\
| . . | | | .....wf .....
| . . | | | 1. i : Id
| . . | | | 2. k : Knd
| . . | | | 3. l : IdLnk
| . . | | | 4. ds : x:Id fp-> Type
| . . | | | 5. da : a:Knd fp-> Type
| . . | | | 6. f :
| . . | | | |  (tg:Id
| . . | | | |   ×State(ds)→ma-valtype(da; k)→
| . . | | | |   (da(rcv(l; tg))?Void List)) List
| . . | | | 7. source(l) = i
| . . | | | 8. D' : Dsys
| . . | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | | 10. w : World
| . . | | | 11. p : FairFifo
| . . | | | 12. PossibleWorld(D';w)
| . . | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 14. ∀e:E.
| . . | | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 15. ∀e:E.
| . . | | | |   isrcv(e)
| . . | | | |   ⇒
| . . | | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | | 17. ∀e:E.
| . . | | | |   loc(e) = i
| . . | | | |   ⇒
| . . | | | |   kind(e) = k
| . . | | | |   ⇒
| . . | | | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
```

```
| . . | | | 19. ∀e:E.
| . . | | | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 20. ∀e:E.
| . . | | | |   isrcv(e)
| . . | | | |   ⇒
| . . | | | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 21. e : E
| . . | | | 22. loc(e) = i
| . . | | | 23. kind(e) = k
| . . | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 27. L :
| . . | | | |   ∀n:ℕ||sends(l;e)||.
| . . | | | |   ∃e':E.
| . . | | | |   isrcv(e')
| . . | | | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | 28. LL :
| . . | | | |   n:ℕ||sends(l;e)||→{e':E
| . . | | | |                      | isrcv(e')
| . . | | | |                        & lnk(e') = l
| . . | | | |                          & sender(e') = e
| . . | | | |                          & index(e') = n }
| . . | | | 29. (λn.1of(L(n))) = LL
| . . | | | 30. ∀e:E.
| . . | | | |   isrcv(e)
| . . | | | |   ⇒
| . . | | | |   sends(lnk(e);sender(e))[index(e)]
| . . | | | |   =
| . . | | | |   msg(lnk(e);tag(e);val(e))
| . . | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | | |   ∃e':E.
| . . | | | |   isrcv(e')
| . . | | | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | ⊢  map(LL;upto(||sends(l;e)||))
| . . | | | |  ∈ {e':E
| . . | | | |      | isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | | | | by Auto
| . . | | | |
| . . | | | .....subterm .....
| . . | | | 1. i : Id
| . . | | | 2. k : Knd
| . . | | | 3. l : IdLnk
| . . | | | 4. ds : x:Id fp-> Type
| . . | | | 5. da : a:Knd fp-> Type
| . . | | | 6. f :
| . . | | |    (tg:Id
| . . | | |     ×State(ds)→ma-valtype(da; k)→
| . . | | |     (da(rcv(l; tg))?Void List)) List
| . . | | | 7. source(l) = i
| . . | | | 8. D' : Dsys
| . . | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | | 10. w : World
| . . | | | 11. p : FairFifo
| . . | | | 12. PossibleWorld(D';w)
```

```
| . . | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 14. ∀e:E.
| . . | | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 15. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | | | 17. ∀e:E.
| . . | | |     loc(e) = i
| . . | | |     ⇒
| . . | | |     kind(e) = k
| . . | | |     ⇒
| . . | | |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 19. ∀e:E.
| . . | | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 20. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 21. e : E
| . . | | | 22. loc(e) = i
| . . | | | 23. kind(e) = k
| . . | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 27. L :
| . . | | |     ∀n:ℕ||sends(l;e)||.
| . . | | |     ∃e':E.
| . . | | |     isrcv(e')
| . . | | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | 28. LL :
| . . | | |     n:ℕ||sends(l;e)||→{e':E
| . . | | |                         | isrcv(e')
| . . | | |                           & lnk(e') = l
| . . | | |                             & sender(e') = e
| . . | | |                             & index(e') = n }
| . . | | | 29. (λn.1of(L(n))) = LL
| . . | | | 30. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     sends(lnk(e);sender(e))[index(e)]
| . . | | |     =
| . . | | |     msg(lnk(e);tag(e);val(e))
| . . | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | |     ∃e':E.
| . . | | |     isrcv(e')
| . . | | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | ⊢  LL
| . . | | |    ∈ ℕ||sends(l;e)||→{e':E
| . . | | |                        | isrcv(e')
| . . | | |                          & lnk(e') = l & sender(e') = e }
| . . | | |   by Ext2 THEN DoSubsume THEN Analyze 0 THEN Analyze -1
```

```
| . . | | |      THEN
| . . | | |      Analyze
| . . | | |      THEN
| . . | | |      ExRepD
| . . | | |
| . . | | |\
| . . | | | .....wf .....
| . . | | | 1. i : Id
| . . | | | 2. k : Knd
| . . | | | 3. l : IdLnk
| . . | | | 4. ds : x:Id fp-> Type
| . . | | | 5. da : a:Knd fp-> Type
| . . | | | 6. f :
| . . | | |    (tg:Id
| . . | | |     ×State(ds)→ma-valtype(da; k)→
| . . | | |     (da(rcv(l; tg))?Void List)) List
| . . | | | 7. source(l) = i
| . . | | | 8. D' : Dsys
| . . | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | | 10. w : World
| . . | | | 11. p : FairFifo
| . . | | | 12. PossibleWorld(D';w)
| . . | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 14. ∀e:E.
| . . | | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 15. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | | | 17. ∀e:E.
| . . | | |     loc(e) = i
| . . | | |     ⇒
| . . | | |     kind(e) = k
| . . | | |     ⇒
| . . | | |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | | 19. ∀e:E.
| . . | | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 20. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | | 21. e : E
| . . | | | 22. loc(e) = i
| . . | | | 23. kind(e) = k
| . . | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | | 27. L :
| . . | | |     ∀n:ℕ||sends(l;e)||.
| . . | | |     ∃e':E.
| . . | | |     isrcv(e')
| . . | | |     & lnk(e') = l & sender(e') = e & index(e') = n
```

```
| . . | | | 28. LL :
| . . | | |     n:ℕ||sends(l;e)||→{e':E
| . . | | |                          | isrcv(e')
| . . | | |                            & lnk(e') = l
| . . | | |                              & sender(e') = e
| . . | | |                              & index(e') = n }
| . . | | | 29. (λn.1of(L(n))) = LL
| . . | | | 30. ∀e:E.
| . . | | |     isrcv(e)
| . . | | |     ⇒
| . . | | |     sends(lnk(e);sender(e))[index(e)]
| . . | | |     =
| . . | | |     msg(lnk(e);tag(e);val(e))
| . . | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | |     ∃e':E.
| . . | | |     isrcv(e')
| . . | | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | | ⊢  {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | | |    ∈ Type  by Auto
| . . | | |
| . . | | .\
| . . | | . 1. i : Id
| . . | | . 2. k : Knd
| . . | | . 3. l : IdLnk
| . . | | . 4. ds : x:Id fp-> Type
| . . | | . 5. da : a:Knd fp-> Type
| . . | | . 6. f :
| . . | | . |  (tg:Id
| . . | | . |   ×State(ds)→ma-valtype(da; k)→
| . . | | . |   (da(rcv(l; tg))?Void List)) List
| . . | | . 7. source(l) = i
| . . | | . 8. D' : Dsys
| . . | | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | . 10. w : World
| . . | | . 11. p : FairFifo
| . . | | . 12. PossibleWorld(D';w)
| . . | | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 14. ∀e:E.
| . . | | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 15. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | . 17. ∀e:E.
| . . | | . |   loc(e) = i

| . . | | . |   ⇒
| . . | | . |   kind(e) = k
| . . | | . |   ⇒
| . . | | . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 19. ∀e:E.
| . . | | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 20. ∀e:E.
```

```
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 21. e : E
| . . | | . 22. loc(e) = i
| . . | | . 23. kind(e) = k
| . . | | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 27. L :
| . . | | . |   ∀n:ℕ||sends(l;e)||.
| . . | | . |   ∃e':E.
| . . | | . |   isrcv(e')
| . . | | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 28. LL :
| . . | | . |   n:ℕ||sends(l;e)||→{e':E
| . . | | . |                     | isrcv(e')
| . . | | . |                       & lnk(e') = l
| . . | | . |                         & sender(e') = e
| . . | | . |                         & index(e') = n }
| . . | | . 29. (λn.1of(L(n))) = LL
| . . | | . 30. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   sends(lnk(e);sender(e))[index(e)]
| . . | | . |   =
| . . | | . |   msg(lnk(e);tag(e);val(e))
| . . | | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | . |   ∃e':E.
| . . | | . |   isrcv(e')
| . . | | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 32. X :
| . . | | . |   {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | | . 33. map(LL;upto(||sends(l;e)||)) = X
| . . | | . ⊢  ||map(λe'.<tag(e'),val(e')>;X)|| = ||sends(l;e)||
| . . | | . | by RWO
| . . | | . |    Thm* ∀f:Top, T:Type, L:T List. ||map(f;L)|| ∼ ||L||
| . . | | . |    {length_DASH_map}
| . . | | . |    0
| . . | | . |
| . . | | . 1. i : Id
| . . | | . 2. k : Knd
| . . | | . 3. l : IdLnk
| . . | | . 4. ds : x:Id fp-> Type
| . . | | . 5. da : a:Knd fp-> Type
| . . | | . 6. f :
| . . | | . |  (tg:Id
| . . | | . |   ×State(ds)→ma-valtype(da; k)→
| . . | | . |   (da(rcv(l; tg))?Void List)) List
| . . | | . 7. source(l) = i
| . . | | . 8. D' : Dsys
| . . | | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | . 10. w : World
| . . | | . 11. p : FairFifo
| . . | | . 12. PossibleWorld(D';w)
```

```
| . . | | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 14. ∀e:E.
| . . | | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 15. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | . 17. ∀e:E.
| . . | | . |   loc(e) = i
| . . | | . |   ⇒
| . . | | . |   kind(e) = k
| . . | | . |   ⇒
| . . | | . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 19. ∀e:E.
| . . | | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 20. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 21. e : E
| . . | | . 22. loc(e) = i
| . . | | . 23. kind(e) = k
| . . | | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 27. L :
| . . | | . |   ∀n:ℕ||sends(l;e)||.
| . . | | . |   ∃e':E.
| . . | | . |   isrcv(e')
| . . | | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 28. LL :
| . . | | . |   n:ℕ||sends(l;e)||→{e':E
| . . | | . |                        | isrcv(e')
| . . | | . |                          & lnk(e') = l
| . . | | . |                            & sender(e') = e
| . . | | . |                            & index(e') = n }
| . . | | . 29. (λn.1of(L(n))) = LL
| . . | | . 30. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   sends(lnk(e);sender(e))[index(e)]
| . . | | . |   =
| . . | | . |   msg(lnk(e);tag(e);val(e))
| . . | | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | . |   ∃e':E.
| . . | | . |   isrcv(e')
| . . | | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 32. X :
| . . | | . |   {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | | . 33. map(LL;upto(||sends(l;e)||)) = X
| . . | | . ⊢  ||X|| = ||sends(l;e)||  by RevHypSubst -1 0
| . . | | . |
```

```
| . . | | . 1. i : Id
| . . | | . 2. k : Knd
| . . | | . 3. l : IdLnk
| . . | | . 4. ds : x:Id fp-> Type
| . . | | . 5. da : a:Knd fp-> Type
| . . | | . 6. f :
| . . | | . |  (tg:Id
| . . | | . |   ×State(ds)→ma-valtype(da; k)→
| . . | | . |   (da(rcv(l; tg))?Void List)) List
| . . | | . 7. source(l) = i
| . . | | . 8. D’ : Dsys
| . . | | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D’
| . . | | . 10. w : World
| . . | | . 11. p : FairFifo
| . . | | . 12. PossibleWorld(D’;w)
| . . | | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 14. ∀e:E.
| . . | | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 15. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | | . 17. ∀e:E.
| . . | | . |   loc(e) = i
| . . | | . |   ⇒
| . . | | . |   kind(e) = k
| . . | | . |   ⇒
| . . | | . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 19. ∀e:E.
| . . | | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 20. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 21. e : E
| . . | | . 22. loc(e) = i
| . . | | . 23. kind(e) = k
| . . | | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 27. L :
| . . | | . |   ∀n:ℕ||sends(l;e)||.
| . . | | . |   ∃e':E.
| . . | | . |   isrcv(e')
| . . | | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 28. LL :
| . . | | . |   n:ℕ||sends(l;e)||→{e':E
| . . | | . |                      | isrcv(e')
| . . | | . |                        & lnk(e') = l
| . . | | . |                          & sender(e') = e
| . . | | . |                          & index(e') = n }
| . . | | . 29. (λn.1of(L(n))) = LL
```

```
| . . | | . 30. ∀e:E.
| . . | | . |   isrcv(e)
| . . | | . |   ⇒
| . . | | . |   sends(lnk(e);sender(e))[index(e)]
| . . | | . |   =
| . . | | . |   msg(lnk(e);tag(e);val(e))
| . . | | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | . |   ∃e':E.
| . . | | . |   isrcv(e')
| . . | | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 32. X :
| . . | | . |   {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | | . 33. map(LL;upto(||sends(l;e)||)) = X
| . . | | . ⊢  ||map(LL;upto(||sends(l;e)||))|| = ||sends(l;e)||
| . . | | . | by RWO
| . . | | . |    Thm* ∀f:Top, T:Type, L:T List. ||map(f;L)|| ~ ||L||
| . . | | . |    {length_DASH_map}
| . . | | . |    0
| . . | | . |
| . . | | . 1. i : Id
| . . | | . 2. k : Knd
| . . | | . 3. l : IdLnk
| . . | | . 4. ds : x:Id fp-> Type
| . . | | . 5. da : a:Knd fp-> Type
| . . | | . 6. f :
| . . | | .    (tg:Id
| . . | | .     ×State(ds)→ma-valtype(da; k)→

| . . | | .     (da(rcv(l; tg))?Void List)) List
| . . | | . 7. source(l) = i
| . . | | . 8. D' : Dsys
| . . | | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | | . 10. w : World
| . . | | . 11. p : FairFifo
| . . | | . 12. PossibleWorld(D';w)
| . . | | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 14. ∀e:E.
| . . | | .     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 15. ∀e:E.
| . . | | .     isrcv(e)
| . . | | .     ⇒
| . . | | .     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | | .     ((λl,tg. da(rcv(l; tg))?Top))
| . . | | . 17. ∀e:E.
| . . | | .     loc(e) = i
| . . | | .     ⇒
| . . | | .     kind(e) = k
| . . | | .     ⇒
| . . | | .     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | | . 19. ∀e:E.
| . . | | .     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 20. ∀e:E.
| . . | | .     isrcv(e)
| . . | | .     ⇒
```

```
| . . | | .     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | | . 21. e : E
| . . | | . 22. loc(e) = i
| . . | | . 23. kind(e) = k
| . . | | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | | . 27. L :
| . . | | .     ∀n:ℕ||sends(l;e)||.
| . . | | .     ∃e':E.
| . . | | .     isrcv(e')
| . . | | .     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 28. LL :
| . . | | .     n:ℕ||sends(l;e)||→{e':E
| . . | | .                           | isrcv(e')
| . . | | .                             & lnk(e') = l
| . . | | .                               & sender(e') = e
| . . | | .                               & index(e') = n }
| . . | | . 29. (λn.1of(L(n))) = LL
| . . | | . 30. ∀e:E.
| . . | | .     isrcv(e)
| . . | | .     ⇒
| . . | | .     sends(lnk(e);sender(e))[index(e)]
| . . | | .     =
| . . | | .     msg(lnk(e);tag(e);val(e))
| . . | | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | | .     ∃e':E.
| . . | | .     isrcv(e')
| . . | | .     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | | . 32. X :
| . . | | .     {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | | . 33. map(LL;upto(||sends(l;e)||)) = X
| . . | | . ⊢  ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | | .   by RWO Thm* ∀n:ℕ. ||upto(n)|| ~ n    {length_upto} 0
| . . | |
| . . | .\
| . . | . 1. i : Id
| . . | . 2. k : Knd
| . . | . 3. l : IdLnk
| . . | . 4. ds : x:Id fp-> Type
| . . | . 5. da : a:Knd fp-> Type
| . . | . 6. f :
| . . | . |  (tg:Id
| . . | . |   ×State(ds)→ma-valtype(da; k)→
| . . | . |   (da(rcv(l; tg))?Void List)) List
| . . | . 7. source(l) = i
| . . | . 8. D' : Dsys
| . . | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . 10. w : World
| . . | . 11. p : FairFifo
| . . | . 12. PossibleWorld(D';w)
| . . | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . 14. ∀e:E.
| . . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . 15. ∀e:E.
```

```
| . . | . |   isrcv(e)
| . . | . |   ⇒
| . . | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . 17. ∀e:E.
| . . | . |   loc(e) = i
| . . | . |   ⇒
| . . | . |   kind(e) = k
| . . | . |   ⇒
| . . | . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . 19. ∀e:E.
| . . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . 20. ∀e:E.
| . . | . |   isrcv(e)
| . . | . |   ⇒
| . . | . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . 21. e : E
| . . | . 22. loc(e) = i
| . . | . 23. kind(e) = k
| . . | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . 27. L :
| . . | . |   ∀n:ℕ||sends(l;e)||.
| . . | . |   ∃e':E.
| . . | . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . 28. LL :
| . . | . |   n:ℕ||sends(l;e)||→{e':E
| . . | . |                        | isrcv(e')
| . . | . |                          & lnk(e') = l
| . . | . |                            & sender(e') = e
| . . | . |                            & index(e') = n }
| . . | . 29. (λn.1of(L(n))) = LL
| . . | . 30. ∀e:E.
| . . | . |   isrcv(e)
| . . | . |   ⇒
| . . | . |   sends(lnk(e);sender(e))[index(e)]
| . . | . |   =
| . . | . |   msg(lnk(e);tag(e);val(e))
| . . | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . |   ∃e':E.
| . . | . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . ⊢  ∀i:ℕ.
| . . | . |  i<||map(λe'.
| . . | . |  <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . |  ⇒
| . . | . |  map(λe'.
| . . | . |  <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))[i]
| . . | . |  =
| . . | . |  map(λm.2of(m);sends(l;e))[i]
| . . | . | by Assert
| . . | . |    ((Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . |                             ×ma-valtype(da; rcv(l; tg)))))
```

```
| . . | . |
| . . | . |\
| . . | . | .....assertion .....
| . . | . | 1. i : Id
| . . | . | 2. k : Knd
| . . | . | 3. l : IdLnk
| . . | . | 4. ds : x:Id fp-> Type
| . . | . | 5. da : a:Knd fp-> Type
| . . | . | 6. f :
| . . | . | |  (tg:Id
| . . | . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . | |   (da(rcv(l; tg))?Void List)) List
| . . | . | 7. source(l) = i
| . . | . | 8. D' : Dsys
| . . | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . | 10. w : World
| . . | . | 11. p : FairFifo
| . . | . | 12. PossibleWorld(D';w)
| . . | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 14. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 15. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . | 17. ∀e:E.
| . . | . | |   loc(e) = i
| . . | . | |   ⇒
| . . | . | |   kind(e) = k
| . . | . | |   ⇒
| . . | . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | 19. ∀e:E.
| . . | . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 20. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | 21. e : E
| . . | . | 22. loc(e) = i
| . . | . | 23. kind(e) = k
| . . | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | 27. L :
| . . | . | |   ∀n:ℕ||sends(l;e)||.
| . . | . | |   ∃e':E.
| . . | . | |   isrcv(e')
| . . | . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | 28. LL :
| . . | . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . | |                     | isrcv(e')
| . . | . | |                       & lnk(e') = l
```

```
| . . | . | |                           & sender(e') = e
| . . | . | |                           & index(e') = n }
| . . | . | 29. (λn.1of(L(n))) = LL
| . . | . | 30. ∀e:E.
| . . | . | |   isrcv(e)
| . . | . | |   ⇒
| . . | . | |   sends(lnk(e);sender(e))[index(e)]
| . . | . | |   =
| . . | . | |   msg(lnk(e);tag(e);val(e))
| . . | . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . | |   ∃e':E.
| . . | . | |   isrcv(e')
| . . | . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | ⊢  (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . | |                        ×ma-valtype(da; rcv(l; tg))))
| . . | . | | by Analyze 0 THEN Analyze -1
| . . | . | |    THEN
| . . | . | |    Assert (x ∈ Msg((λl,tg. da(rcv(l; tg))?Top)))
| . . | . | |
| . . | . | |\
| . . | . | | .....assertion .....
| . . | . | | 1. i : Id
| . . | . | | 2. k : Knd
| . . | . | | 3. l : IdLnk
| . . | . | | 4. ds : x:Id fp-> Type
| . . | . | | 5. da : a:Knd fp-> Type
| . . | . | | 6. f :
| . . | . | | |  (tg:Id
| . . | . | | |   ×State(ds)→ma-valtype(da; k)→
| . . | . | | |   (da(rcv(l; tg))?Void List)) List
| . . | . | | 7. source(l) = i
| . . | . | | 8. D' : Dsys
| . . | . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . | | 10. w : World
| . . | . | | 11. p : FairFifo
| . . | . | | 12. PossibleWorld(D';w)
| . . | . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | | 14. ∀e:E.
| . . | . | | |   loc(e) = i
| . . | . | | |   ⇒
| . . | . | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | | 15. ∀e:E.
| . . | . | | |   isrcv(e)
| . . | . | | |   ⇒
| . . | . | | |   lnk(e) = l
| . . | . | | |   ⇒
| . . | . | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . | | 17. ∀e:E.
| . . | . | | |   loc(e) = i
| . . | . | | |   ⇒
| . . | . | | |   kind(e) = k
| . . | . | | |   ⇒
| . . | . | | |   sends(l;e)
```

```
| . . | . | | |   =
| . . | . | | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | | 19. ∀e:E.
| . . | . | | |   loc(e) = i
| . . | . | | |   ⇒
| . . | . | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | | 20. ∀e:E.
| . . | . | | |   isrcv(e)
| . . | . | | |   ⇒
| . . | . | | |   lnk(e) = l
| . . | . | | |   ⇒
| . . | . | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | | 21. e : E
| . . | . | | 22. loc(e) = i
| . . | . | | 23. kind(e) = k
| . . | . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | | 25. sends(l;e)
| . . | . | | |   =
| . . | . | | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | | 27. L :
| . . | . | | |   ∀n:ℕ||sends(l;e)||.
| . . | . | | |   ∃e':E.
| . . | . | | |   isrcv(e')
| . . | . | | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | | 28. LL :
| . . | . | | |   n:ℕ||sends(l;e)||→{e':E
| . . | . | | |                       | isrcv(e')
| . . | . | | |                         & lnk(e') = l
| . . | . | | |                           & sender(e') = e
| . . | . | | |                           & index(e') = n }
| . . | . | | 29. (λn.1of(L(n))) = LL
| . . | . | | 30. ∀e:E.
| . . | . | | |   isrcv(e)
| . . | . | | |   ⇒
| . . | . | | |   sends(lnk(e);sender(e))[index(e)]
| . . | . | | |   =
| . . | . | | |   msg(lnk(e);tag(e);val(e))
| . . | . | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . | | |   ∃e':E.
| . . | . | | |   isrcv(e')
| . . | . | | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | | 32. x : Msg
| . . | . | | 33. haslink(l; x)
| . . | . | | ⊢  x ∈ Msg((λl,tg. da(rcv(l; tg))?Top))
| . . | . | | | by Assert (x ∈ {m:Msg| source(mlnk(m)) = i })
| . . | . | | |    THEN
| . . | . | | |    TrySubsume
| . . | . | | |    THEN
| . . | . | | |    Analyze
| . . | . | | |

| . . | . | | .....set predicate .....
| . . | . | | 1. i : Id
| . . | . | | 2. k : Knd
```

```
| . . | . | | 3. l : IdLnk
| . . | . | | 4. ds : x:Id fp-> Type
| . . | . | | 5. da : a:Knd fp-> Type
| . . | . | | 6. f :
| . . | . | |    (tg:Id
| . . | . | |     ×State(ds)→ma-valtype(da; k)→
| . . | . | |     (da(rcv(l; tg))?Void List)) List
| . . | . | | 7. source(l) = i
| . . | . | | 8. D' : Dsys
| . . | . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . | | 10. w : World
| . . | . | | 11. p : FairFifo
| . . | . | | 12. PossibleWorld(D';w)
| . . | . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | | 14. ∀e:E.
| . . | . | |     loc(e) = i
| . . | . | |     ⇒
| . . | . | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | | 15. ∀e:E.
| . . | . | |     isrcv(e)
| . . | . | |     ⇒
| . . | . | |     lnk(e) = l
| . . | . | |     ⇒
| . . | . | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . | |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . | | 17. ∀e:E.
| . . | . | |     loc(e) = i
| . . | . | |     ⇒
| . . | . | |     kind(e) = k
| . . | . | |     ⇒
| . . | . | |     sends(l;e)
| . . | . | |     =
| . . | . | |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | | 19. ∀e:E.
| . . | . | |     loc(e) = i
| . . | . | |     ⇒
| . . | . | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | | 20. ∀e:E.
| . . | . | |     isrcv(e)
| . . | . | |     ⇒
| . . | . | |     lnk(e) = l
| . . | . | |     ⇒
| . . | . | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | | 21. e : E
| . . | . | | 22. loc(e) = i
| . . | . | | 23. kind(e) = k
| . . | . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | | 25. sends(l;e)
| . . | . | |     =
| . . | . | |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | | 27. L :
| . . | . | |     ∀n:ℕ||sends(l;e)||.
```

```
| . . | . | |     ∃e':E.
| . . | . | |     isrcv(e')
| . . | . | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | | 28. LL :
| . . | . | |     n:ℕ||sends(l;e)||→{e':E
| . . | . | |                         | isrcv(e')
| . . | . | |                           & lnk(e') = l
| . . | . | |                             & sender(e') = e
| . . | . | |                             & index(e') = n }
| . . | . | | 29. (λn.1of(L(n))) = LL
| . . | . | | 30. ∀e:E.
| . . | . | |     isrcv(e)
| . . | . | |     ⇒
| . . | . | |     sends(lnk(e);sender(e))[index(e)]
| . . | . | |     =
| . . | . | |     msg(lnk(e);tag(e);val(e))
| . . | . | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . | |     ∃e':E.
| . . | . | |     isrcv(e')
| . . | . | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | | 32. x : Msg
| . . | . | | 33. haslink(l; x)
| . . | . | | ⊢  source(mlnk(x)) = i
| . . | . | |   by Unfold `haslink` -1 THEN HypSubst' -1 0
| . . | . | |
| . . | . | .\
| . . | . | . 1. i : Id
| . . | . | . 2. k : Knd
| . . | . | . 3. l : IdLnk
| . . | . | . 4. ds : x:Id fp-> Type
| . . | . | . 5. da : a:Knd fp-> Type
| . . | . | . 6. f :
| . . | . | .    (tg:Id
| . . | . | .     ×State(ds)→ma-valtype(da; k)→
| . . | . | .     (da(rcv(l; tg))?Void List)) List
| . . | . | . 7. source(l) = i
| . . | . | . 8. D' : Dsys
| . . | . | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . | . 10. w : World
| . . | . | . 11. p : FairFifo
| . . | . | . 12. PossibleWorld(D';w)
| . . | . | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | . 14. ∀e:E.
| . . | . | .     loc(e) = i
| . . | . | .     ⇒
| . . | . | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | . 15. ∀e:E.
| . . | . | .     isrcv(e)
| . . | . | .     ⇒
| . . | . | .     lnk(e) = l
| . . | . | .     ⇒
| . . | . | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . | .     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . | . 17. ∀e:E.
```

```
| . . | . | .     loc(e) = i
| . . | . | .     ⇒
| . . | . | .     kind(e) = k
| . . | . | .     ⇒
| . . | . | .     sends(l;e)
| . . | . | .     =
| . . | . | .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . | . 19. ∀e:E.
| . . | . | .     loc(e) = i
| . . | . | .     ⇒
| . . | . | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | . 20. ∀e:E.
| . . | . | .     isrcv(e)
| . . | . | .     ⇒
| . . | . | .     lnk(e) = l
| . . | . | .     ⇒
| . . | . | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . | . 21. e : E
| . . | . | . 22. loc(e) = i
| . . | . | . 23. kind(e) = k
| . . | . | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | . 25. sends(l;e)
| . . | . | .     =
| . . | . | .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . | . 27. L :
| . . | . | .     ∀n:ℕ||sends(l;e)||.
| . . | . | .     ∃e':E.
| . . | . | .     isrcv(e')
| . . | . | .     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | . 28. LL :
| . . | . | .     n:ℕ||sends(l;e)||→{e':E
| . . | . | .                          | isrcv(e')
| . . | . | .                            & lnk(e') = l
| . . | . | .                              & sender(e') = e
| . . | . | .                              & index(e') = n }
| . . | . | . 29. (λn.1of(L(n))) = LL
| . . | . | . 30. ∀e:E.
| . . | . | .     isrcv(e)
| . . | . | .     ⇒
| . . | . | .     sends(lnk(e);sender(e))[index(e)]
| . . | . | .     =
| . . | . | .     msg(lnk(e);tag(e);val(e))
| . . | . | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . | .     ∃e':E.
| . . | . | .     isrcv(e')
| . . | . | .     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . | . 32. x : Msg
| . . | . | . 33. haslink(l; x)
| . . | . | . 34. x ∈ Msg((λl,tg. da(rcv(l; tg))?Top))
| . . | . | . ⊢  x ∈ IdLnk×(tg:Id×ma-valtype(da; rcv(l; tg)))
| . . | . | .   by MoveToConcl -2 THEN Unfold `haslink` 0
| . . | . | .      THEN
| . . | . | .      GenConclAtAddr [2;2]
```

```
| . . | . | .      THEN
| . . | . | .      Thin -1
| . . | . | .      THEN
| . . | . | .      Unfolds [‘mlnk‘;‘ma-valtype‘] 0
| . . | . | .      THEN
| . . | . | .      Analyze -1
| . . | . | .      THEN
| . . | . | .      All Reduce
| . . | . | .      THEN
| . . | . | .      Analyze 0
| . . | . | .      THEN
| . . | . | .      DoSubsume
| . . | . | .      THEN
| . . | . | .      HypSubst' -3 0
| . . | . |
| . . | . .\
| . . | . . 1. i : Id
| . . | . . 2. k : Knd
| . . | . . 3. l : IdLnk
| . . | . . 4. ds : x:Id fp-> Type
| . . | . . 5. da : a:Knd fp-> Type
| . . | . . 6. f :
| . . | . . |  (tg:Id
| . . | . . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . |   (da(rcv(l; tg))?Void List)) List
| . . | . . 7. source(l) = i
| . . | . . 8. D' : Dsys
| . . | . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . 10. w : World
| . . | . . 11. p : FairFifo
| . . | . . 12. PossibleWorld(D';w)
| . . | . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . 14. ∀e:E.
| . . | . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . 15. ∀e:E.
| . . | . . |   isrcv(e)
| . . | . . |   ⇒
| . . | . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . 17. ∀e:E.
| . . | . . |   loc(e) = i
| . . | . . |   ⇒
| . . | . . |   kind(e) = k
| . . | . . |   ⇒
| . . | . . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . 19. ∀e:E.
| . . | . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . 20. ∀e:E.
| . . | . . |   isrcv(e)
| . . | . . |   ⇒
| . . | . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . 21. e : E
| . . | . . 22. loc(e) = i
```

```
| . . | . . 23. kind(e) = k
| . . | . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . 27. L :
| . . | . . |   ∀n:ℕ||sends(l;e)||.
| . . | . . |   ∃e':E.
| . . | . . |   isrcv(e')
| . . | . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . 28. LL :
| . . | . . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . |                      | isrcv(e')
| . . | . . |                        & lnk(e') = l
| . . | . . |                          & sender(e') = e
| . . | . . |                          & index(e') = n }
| . . | . . 29. (λn.1of(L(n))) = LL
| . . | . . 30. ∀e:E.
| . . | . . |   isrcv(e)
| . . | . . |   ⇒
| . . | . . |   sends(lnk(e);sender(e))[index(e)]
| . . | . . |   =
| . . | . . |   msg(lnk(e);tag(e);val(e))
| . . | . . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . |   ∃e':E.
| . . | . . |   isrcv(e')
| . . | . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . |                        ×ma-valtype(da; rcv(l; tg))))
| . . | . . ⊢  ∀i:ℕ.
| . . | . . |  i<||map(λe'.
| . . | . . |  <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . |  ⇒
| . . | . . |  map(λe'.
| . . | . . |  <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))[i]
| . . | . . |  =
| . . | . . |  map(λm.2of(m);sends(l;e))[i]
| . . | . . | by Assert
| . . | . . |    (map(LL;upto(||sends(l;e)||))
| . . | . . |     ∈ {e':E
| . . | . . |        | isrcv(e')
| . . | . . |          & lnk(e') = l & sender(e') = e } List)
| . . | . . |
| . . | . . |\
| . . | . . | .....assertion .....
| . . | . . | 1. i : Id
| . . | . . | 2. k : Knd
| . . | . . | 3. l : IdLnk
| . . | . . | 4. ds : x:Id fp-> Type
| . . | . . | 5. da : a:Knd fp-> Type
| . . | . . | 6. f :
| . . | . . | |  (tg:Id
| . . | . . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . . | |   (da(rcv(l; tg))?Void List)) List
| . . | . . | 7. source(l) = i
| . . | . . | 8. D' : Dsys
```

```
| . . | . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . | 10. w : World
| . . | . . | 11. p : FairFifo
| . . | . . | 12. PossibleWorld(D';w)
| . . | . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . | 14. ∀e:E.
| . . | . . | |   loc(e) = i
| . . | . . | |   ⇒
| . . | . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . | 15. ∀e:E.
| . . | . . | |   isrcv(e)
| . . | . . | |   ⇒
| . . | . . | |   lnk(e) = l
| . . | . . | |   ⇒
| . . | . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . | 17. ∀e:E.
| . . | . . | |   loc(e) = i
| . . | . . | |   ⇒
| . . | . . | |   kind(e) = k
| . . | . . | |   ⇒
| . . | . . | |   sends(l;e)
| . . | . . | |   =
| . . | . . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . | 19. ∀e:E.
| . . | . . | |   loc(e) = i
| . . | . . | |   ⇒
| . . | . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . | 20. ∀e:E.
| . . | . . | |   isrcv(e)
| . . | . . | |   ⇒
| . . | . . | |   lnk(e) = l
| . . | . . | |   ⇒
| . . | . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . | 21. e : E
| . . | . . | 22. loc(e) = i
| . . | . . | 23. kind(e) = k
| . . | . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . | 25. sends(l;e)
| . . | . . | |   =
| . . | . . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . | 27. L :
| . . | . . | |   ∀n:ℕ||sends(l;e)||.
| . . | . . | |   ∃e':E.
| . . | . . | |   isrcv(e')
| . . | . . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . | 28. LL :
| . . | . . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . | |                       | isrcv(e')
| . . | . . | |                         & lnk(e') = l
| . . | . . | |                           & sender(e') = e
| . . | . . | |                           & index(e') = n }
```

```
| . . | . . | 29. (λn.1of(L(n))) = LL
| . . | . . | 30. ∀e:E.
| . . | . . | |   isrcv(e)
| . . | . . | |   ⇒
| . . | . . | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . | |   =
| . . | . . | |   msg(lnk(e);tag(e);val(e))
| . . | . . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . | |   ∃e':E.
| . . | . . | |   isrcv(e')
| . . | . . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . | |                           ×ma-valtype(da; rcv(l; tg))))
| . . | . . | ⊢  map(LL;upto(||sends(l;e)||))
| . . | . . | |  ∈ {e':E
| . . | . . | |     | isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | . . | | by Auto
| . . | . . | |
| . . | . . | .....subterm .....
| . . | . . | 1. i : Id
| . . | . . | 2. k : Knd
| . . | . . | 3. l : IdLnk
| . . | . . | 4. ds : x:Id fp-> Type
| . . | . . | 5. da : a:Knd fp-> Type
| . . | . . | 6. f :
| . . | . . |    (tg:Id
| . . | . . |     ×State(ds)→ma-valtype(da; k)→
| . . | . . |     (da(rcv(l; tg))?Void List)) List
| . . | . . | 7. source(l) = i
| . . | . . | 8. D' : Dsys
| . . | . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . | 10. w : World
| . . | . . | 11. p : FairFifo
| . . | . . | 12. PossibleWorld(D';w)
| . . | . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . | 14. ∀e:E.
| . . | . . |     loc(e) = i
| . . | . . |     ⇒
| . . | . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . | 15. ∀e:E.
| . . | . . |     isrcv(e)
| . . | . . |     ⇒
| . . | . . |     lnk(e) = l
| . . | . . |     ⇒
| . . | . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . | 17. ∀e:E.
| . . | . . |     loc(e) = i
| . . | . . |     ⇒
| . . | . . |     kind(e) = k
| . . | . . |     ⇒
| . . | . . |     sends(l;e)
| . . | . . |     =
| . . | . . |     tagged-messages(l;λz.(z when e);val(e);f)
```

```
| . . | . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . | 19. ∀e:E.
| . . | . . |     loc(e) = i
| . . | . . |     ⇒
| . . | . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . | 20. ∀e:E.
| . . | . . |     isrcv(e)
| . . | . . |     ⇒
| . . | . . |     lnk(e) = l
| . . | . . |     ⇒
| . . | . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . | 21. e : E
| . . | . . | 22. loc(e) = i
| . . | . . | 23. kind(e) = k
| . . | . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . | 25. sends(l;e)
| . . | . . |     =
| . . | . . |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . | 27. L :
| . . | . . |     ∀n:ℕ||sends(l;e)||.
| . . | . . |     ∃e':E.
| . . | . . |     isrcv(e')
| . . | . . |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . | 28. LL :
| . . | . . |     n:ℕ||sends(l;e)||→{e':E
| . . | . . |                         | isrcv(e')
| . . | . . |                           & lnk(e') = l
| . . | . . |                             & sender(e') = e
| . . | . . |                             & index(e') = n }
| . . | . . | 29. (λn.1of(L(n))) = LL
| . . | . . | 30. ∀e:E.
| . . | . . |     isrcv(e)
| . . | . . |     ⇒
| . . | . . |     sends(lnk(e);sender(e))[index(e)]
| . . | . . |     =
| . . | . . |     msg(lnk(e);tag(e);val(e))
| . . | . . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . |     ∃e':E.
| . . | . . |     isrcv(e')
| . . | . . |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . |                           ×ma-valtype(da; rcv(l; tg))))
| . . | . . | ⊢  LL
| . . | . . |    ∈ ℕ||sends(l;e)||→{e':E
| . . | . . |                       | isrcv(e')
| . . | . . |                         & lnk(e') = l
| . . | . . |                           & sender(e') = e }
| . . | . . |   by Ext2 THEN DoSubsume THEN Analyze 0 THEN Analyze -1
| . . | . . |     THEN
| . . | . . |     Analyze
| . . | . . |     THEN
| . . | . . |     ExRepD
| . . | . . |
| . . | . . .\
```

```
| . . | . . . 1. i : Id
| . . | . . . 2. k : Knd
| . . | . . . 3. l : IdLnk
| . . | . . . 4. ds : x:Id fp-> Type
| . . | . . . 5. da : a:Knd fp-> Type
| . . | . . . 6. f :
| . . | . . . |  (tg:Id
| . . | . . . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . |   (da(rcv(l; tg))?Void List)) List
| . . | . . . 7. source(l) = i
| . . | . . . 8. D' : Dsys
| . . | . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . 10. w : World
| . . | . . . 11. p : FairFifo
| . . | . . . 12. PossibleWorld(D';w)
| . . | . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . 14. ∀e:E.
| . . | . . . |   loc(e) = i
| . . | . . . |   ⇒
| . . | . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . 15. ∀e:E.
| . . | . . . |   isrcv(e)
| . . | . . . |   ⇒
| . . | . . . |   lnk(e) = l
| . . | . . . |   ⇒
| . . | . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . 17. ∀e:E.
| . . | . . . |   loc(e) = i
| . . | . . . |   ⇒
| . . | . . . |   kind(e) = k
| . . | . . . |   ⇒
| . . | . . . |   sends(l;e)
| . . | . . . |   =
| . . | . . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . 19. ∀e:E.
| . . | . . . |   loc(e) = i
| . . | . . . |   ⇒
| . . | . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . 20. ∀e:E.
| . . | . . . |   isrcv(e)
| . . | . . . |   ⇒
| . . | . . . |   lnk(e) = l
| . . | . . . |   ⇒
| . . | . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . 21. e : E
| . . | . . . 22. loc(e) = i
| . . | . . . 23. kind(e) = k
| . . | . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . 25. sends(l;e)
| . . | . . . |   =
| . . | . . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
```

```
| . . | . . . 27. L :
| . . | . . . |   ∀n:ℕ||sends(l;e)||.
| . . | . . . |   ∃e':E.
| . . | . . . |   isrcv(e')
| . . | . . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . 28. LL :
| . . | . . . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . |                       | isrcv(e')
| . . | . . . |                         & lnk(e') = l
| . . | . . . |                           & sender(e') = e
| . . | . . . |                           & index(e') = n }
| . . | . . . 29. (λn.1of(L(n))) = LL
| . . | . . . 30. ∀e:E.
| . . | . . . |   isrcv(e)
| . . | . . . |   ⇒
| . . | . . . |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . |   =
| . . | . . . |   msg(lnk(e);tag(e);val(e))
| . . | . . . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . |   ∃e':E.
| . . | . . . |   isrcv(e')
| . . | . . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . |                         ×ma-valtype(da; rcv(l; tg))))
| . . | . . . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . |   ∈ {e':E
| . . | . . . |      | isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | . . . ⊢  ∀i:ℕ.
| . . | . . . |  i<||map(λe'.
| . . | . . . |  <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . . |  ⇒
| . . | . . . |  map(λe'.
| . . | . . . |  <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))[i]
| . . | . . . |  =
| . . | . . . |  map(λm.2of(m);sends(l;e))[i]
| . . | . . . | by AssertBY
| . . | . . . |    (||upto(||sends(l;e)||)|| = ||sends(l;e)||)
| . . | . . . |    (RWO Thm* ∀n:ℕ. ||upto(n)|| ~ n    {length_upto} 0)
| . . | . . . |    THEN
| . . | . . . |    AssertBY
| . . | . . . |    (||map(LL;upto(||sends(l;e)||))|| = ||sends(l;e)||)
| . . | . . . |    (RWO
| . . | . . . |     Thm* ∀f:Top, T:Type, L:T List.
| . . | . . . |          ||map(f;L)|| ~ ||L||
| . . | . . . |     {length_DASH_map}
| . . | . . . |     0)
| . . | . . . |    THEN
| . . | . . . |    AssertBY
| . . | . . . |    (||map(λe'.
| . . | . . . |     <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . . |     =
| . . | . . . |     ||sends(l;e)||)
| . . | . . . |    (RWW
| . . | . . . |     Thm* ∀f:Top, T:Type, L:T List.
| . . | . . . |          ||map(f;L)|| ~ ||L||
```

```
| . . | . . . |     {length_DASH_map}
| . . | . . . |     0)
| . . | . . . |    THEN
| . . | . . . |    Try (Unfold ‘label‘ 0 THEN Trivial)
| . . | . . . |    THEN
| . . | . . . |    RWW
| . . | . . . |    Thm* ∀f:Top, T:Type, L:T List, i:ℕ||L||.
| . . | . . . |         map(f;L)[i] ~ (f(L[i]))
| . . | . . . |    {select_DASH_map}
| . . | . . . |    0
| . . | . . . |    THEN
| . . | . . . |    Reduce 0
| . . | . . . |
| . . | . . . 1. i : Id
| . . | . . . 2. k : Knd
| . . | . . . 3. l : IdLnk
| . . | . . . 4. ds : x:Id fp-> Type
| . . | . . . 5. da : a:Knd fp-> Type
| . . | . . . 6. f :
| . . | . . . |  (tg:Id
| . . | . . . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . |   (da(rcv(l; tg))?Void List)) List
| . . | . . . 7. source(l) = i
| . . | . . . 8. D’ : Dsys
| . . | . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D’
| . . | . . . 10. w : World
| . . | . . . 11. p : FairFifo
| . . | . . . 12. PossibleWorld(D’;w)
| . . | . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . 14. ∀e:E.
| . . | . . . |   loc(e) = i
| . . | . . . |   ⇒
| . . | . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . 15. ∀e:E.
| . . | . . . |   isrcv(e)
| . . | . . . |   ⇒
| . . | . . . |   lnk(e) = l
| . . | . . . |   ⇒
| . . | . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . 17. ∀e:E.
| . . | . . . |   loc(e) = i
| . . | . . . |   ⇒
| . . | . . . |   kind(e) = k
| . . | . . . |   ⇒
| . . | . . . |   sends(l;e)
| . . | . . . |   =
| . . | . . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . 19. ∀e:E.
| . . | . . . |   loc(e) = i
| . . | . . . |   ⇒
| . . | . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . 20. ∀e:E.
```

```
| . . | . . . |   isrcv(e)
| . . | . . . |   ⇒
| . . | . . . |   lnk(e) = l
| . . | . . . |   ⇒
| . . | . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . 21. e : E
| . . | . . . 22. loc(e) = i
| . . | . . . 23. kind(e) = k
| . . | . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . 25. sends(l;e)
| . . | . . . |   =
| . . | . . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . 27. L :
| . . | . . . |   ∀n:ℕ||sends(l;e)||.
| . . | . . . |   ∃e':E.
| . . | . . . |   isrcv(e')
| . . | . . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . 28. LL :
| . . | . . . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . |                      | isrcv(e')
| . . | . . . |                        & lnk(e') = l
| . . | . . . |                          & sender(e') = e
| . . | . . . |                          & index(e') = n }
| . . | . . . 29. (λn.1of(L(n))) = LL
| . . | . . . 30. ∀e:E.
| . . | . . . |   isrcv(e)
| . . | . . . |   ⇒
| . . | . . . |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . |   =
| . . | . . . |   msg(lnk(e);tag(e);val(e))
| . . | . . . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . |   ∃e':E.
| . . | . . . |   isrcv(e')
| . . | . . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . |                        ×ma-valtype(da; rcv(l; tg))))
| . . | . . . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . |   ∈ {e':E
| . . | . . . |      | isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . | . . . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . 35. ||map(LL;upto(||sends(l;e)||))|| = ||sends(l;e)||
| . . | . . . 36. ||map(λe'.
| . . | . . . |   <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . . |   =
| . . | . . . |   ||sends(l;e)||
| . . | . . . 37. i1 : ℕ
| . . | . . . 38. i1<||map(λe'.
| . . | . . . |   <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . . ⊢  <tag(LL(upto(||sends(l;e)||)[i1]))
| . . | . . . |  ,val(LL(upto(||sends(l;e)||)[i1]))>
| . . | . . . |  =
| . . | . . . |  2of(sends(l;e)[i1])
| . . | . . . | by Subst' (upto(||sends(l;e)||)[i1] = i1) 0
| . . | . . . |
```

```
| . . | . . . |\
| . . | . . . | 1. i : Id
| . . | . . . | 2. k : Knd
| . . | . . . | 3. l : IdLnk
| . . | . . . | 4. ds : x:Id fp-> Type
| . . | . . . | 5. da : a:Knd fp-> Type
| . . | . . . | 6. f :
| . . | . . . |    (tg:Id
| . . | . . . |     ×State(ds)→ma-valtype(da; k)→
| . . | . . . |     (da(rcv(l; tg))?Void List)) List
| . . | . . . | 7. source(l) = i
| . . | . . . | 8. D' : Dsys
| . . | . . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . | 10. w : World
| . . | . . . | 11. p : FairFifo
| . . | . . . | 12. PossibleWorld(D';w)
| . . | . . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . | 14. ∀e:E.
| . . | . . . |     loc(e) = i
| . . | . . . |     ⇒
| . . | . . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . | 15. ∀e:E.
| . . | . . . |     isrcv(e)
| . . | . . . |     ⇒
| . . | . . . |     lnk(e) = l
| . . | . . . |     ⇒
| . . | . . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . | 17. ∀e:E.
| . . | . . . |     loc(e) = i
| . . | . . . |     ⇒
| . . | . . . |     kind(e) = k
| . . | . . . |     ⇒
| . . | . . . |     sends(l;e)
| . . | . . . |     =
| . . | . . . |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . | 19. ∀e:E.
| . . | . . . |     loc(e) = i
| . . | . . . |     ⇒
| . . | . . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . | 20. ∀e:E.
| . . | . . . |     isrcv(e)
| . . | . . . |     ⇒
| . . | . . . |     lnk(e) = l
| . . | . . . |     ⇒
| . . | . . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . | 21. e : E
| . . | . . . | 22. loc(e) = i
| . . | . . . | 23. kind(e) = k
| . . | . . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . | 25. sends(l;e)
| . . | . . . |     =
| . . | . . . |     tagged-messages(l;λz.(z when e);val(e);f)
```

```
| . . | . . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . | 27. L :
| . . | . . . |     ∀n:ℕ||sends(l;e)||.
| . . | . . . |     ∃e':E.
| . . | . . . |     isrcv(e')
| . . | . . . |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . | 28. LL :
| . . | . . . |     n:ℕ||sends(l;e)||→{e':E
| . . | . . . |                           | isrcv(e')
| . . | . . . |                             & lnk(e') = l
| . . | . . . |                               & sender(e') = e
| . . | . . . |                               & index(e') = n }
| . . | . . . | 29. (λn.1of(L(n))) = LL
| . . | . . . | 30. ∀e:E.
| . . | . . . |     isrcv(e)
| . . | . . . |     ⇒
| . . | . . . |     sends(lnk(e);sender(e))[index(e)]
| . . | . . . |     =
| . . | . . . |     msg(lnk(e);tag(e);val(e))
| . . | . . . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . |     ∃e':E.
| . . | . . . |     isrcv(e')
| . . | . . . |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . |                             ×ma-valtype(da; rcv(l; tg)
| . . | . . . |                                           )))
| . . | . . . | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . |     ∈ {e':E
| . . | . . . |        | isrcv(e')
| . . | . . . |          & lnk(e') = l & sender(e') = e } List
| . . | . . . | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . | 35. ||map(LL;upto(||sends(l;e)||))|| = ||sends(l;e)||
| . . | . . . | 36. ||map(λe'.
| . . | . . . |     <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . . |     =
| . . | . . . |     ||sends(l;e)||
| . . | . . . | 37. i1 : ℕ
| . . | . . . | 38. i1<||map(λe'.
| . . | . . . |     <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . . | ⊢  upto(||sends(l;e)||)[i1] = i1
| . . | . . . |   by BackThru
| . . | . . . |      Thm* ∀m:ℕ, n:ℕm. upto(m)[n] = n    {select_upto}
| . . | . . . |
| . . | . . . .\
| . . | . . . . 1. i : Id
| . . | . . . . 2. k : Knd
| . . | . . . . 3. l : IdLnk
| . . | . . . . 4. ds : x:Id fp-> Type
| . . | . . . . 5. da : a:Knd fp-> Type
| . . | . . . . 6. f :
| . . | . . . . |  (tg:Id
| . . | . . . . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . 7. source(l) = i
| . . | . . . . 8. D' : Dsys
```

```
| . . | . . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . 10. w : World
| . . | . . . . 11. p : FairFifo
| . . | . . . . 12. PossibleWorld(D';w)
| . . | . . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . 14. ∀e:E.
| . . | . . . . |   loc(e) = i
| . . | . . . . |   ⇒
| . . | . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . 15. ∀e:E.
| . . | . . . . |   isrcv(e)
| . . | . . . . |   ⇒
| . . | . . . . |   lnk(e) = l
| . . | . . . . |   ⇒
| . . | . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . 17. ∀e:E.
| . . | . . . . |   loc(e) = i
| . . | . . . . |   ⇒
| . . | . . . . |   kind(e) = k
| . . | . . . . |   ⇒
| . . | . . . . |   sends(l;e)
| . . | . . . . |   =
| . . | . . . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . 19. ∀e:E.

| . . | . . . . |   loc(e) = i
| . . | . . . . |   ⇒
| . . | . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . 20. ∀e:E.
| . . | . . . . |   isrcv(e)
| . . | . . . . |   ⇒
| . . | . . . . |   lnk(e) = l
| . . | . . . . |   ⇒
| . . | . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . 21. e : E
| . . | . . . . 22. loc(e) = i
| . . | . . . . 23. kind(e) = k
| . . | . . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . 25. sends(l;e)
| . . | . . . . |   =
| . . | . . . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . 27. L :
| . . | . . . . |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . |   ∃e':E.
| . . | . . . . |   isrcv(e')
| . . | . . . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . 28. LL :
| . . | . . . . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . |                        | isrcv(e')
| . . | . . . . |                          & lnk(e') = l
| . . | . . . . |                            & sender(e') = e
| . . | . . . . |                            & index(e') = n }
```

```
| . . | . . . . 29. (λn.1of(L(n))) = LL
| . . | . . . . 30. ∀e:E.
| . . | . . . . |   isrcv(e)
| . . | . . . . |   ⇒
| . . | . . . . |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . |   =
| . . | . . . . |   msg(lnk(e);tag(e);val(e))
| . . | . . . . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . |   ∃e':E.
| . . | . . . . |   isrcv(e')
| . . | . . . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . |                         ×ma-valtype(da; rcv(l; tg)
| . . | . . . . |                                     )))
| . . | . . . . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . |   ∈ {e':E
| . . | . . . . |      | isrcv(e')
| . . | . . . . |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . 35. ||map(LL;upto(||sends(l;e)||))|| = ||sends(l;e)||
| . . | . . . . 36. ||map(λe'.
| . . | . . . . |   <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . . . |   =
| . . | . . . . |   ||sends(l;e)||
| . . | . . . . 37. i1 : ℕ
| . . | . . . . 38. i1<||map(λe'.
| . . | . . . . |   <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . . . ⊢  <tag(LL(i1)),val(LL(i1))> = 2of(sends(l;e)[i1])
| . . | . . . . | by GenConclAtAddr [2;1;2]
| . . | . . . . |
| . . | . . . . 1. i : Id
| . . | . . . . 2. k : Knd
| . . | . . . . 3. l : IdLnk
| . . | . . . . 4. ds : x:Id fp-> Type
| . . | . . . . 5. da : a:Knd fp-> Type
| . . | . . . . 6. f :
| . . | . . . . |  (tg:Id
| . . | . . . . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . 7. source(l) = i
| . . | . . . . 8. D' : Dsys
| . . | . . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . 10. w : World
| . . | . . . . 11. p : FairFifo
| . . | . . . . 12. PossibleWorld(D';w)
| . . | . . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . 14. ∀e:E.
| . . | . . . . |   loc(e) = i
| . . | . . . . |   ⇒
| . . | . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . 15. ∀e:E.
| . . | . . . . |   isrcv(e)
| . . | . . . . |   ⇒
| . . | . . . . |   lnk(e) = l
| . . | . . . . |   ⇒
```

```
| . . | . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . 17. ∀e:E.
| . . | . . . . |   loc(e) = i
| . . | . . . . |   ⇒
| . . | . . . . |   kind(e) = k
| . . | . . . . |   ⇒
| . . | . . . . |   sends(l;e)
| . . | . . . . |   =
| . . | . . . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . 19. ∀e:E.
| . . | . . . . |   loc(e) = i
| . . | . . . . |   ⇒
| . . | . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . 20. ∀e:E.
| . . | . . . . |   isrcv(e)
| . . | . . . . |   ⇒
| . . | . . . . |   lnk(e) = l
| . . | . . . . |   ⇒
| . . | . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . 21. e : E
| . . | . . . . 22. loc(e) = i
| . . | . . . . 23. kind(e) = k
| . . | . . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . 25. sends(l;e)
| . . | . . . . |   =
| . . | . . . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . 27. L :
| . . | . . . . |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . |   ∃e':E.
| . . | . . . . |   isrcv(e')
| . . | . . . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . 28. LL :
| . . | . . . . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . |                     | isrcv(e')
| . . | . . . . |                       & lnk(e') = l
| . . | . . . . |                         & sender(e') = e
| . . | . . . . |                         & index(e') = n }
| . . | . . . . 29. (λn.1of(L(n))) = LL
| . . | . . . . 30. ∀e:E.
| . . | . . . . |   isrcv(e)
| . . | . . . . |   ⇒
| . . | . . . . |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . |   =
| . . | . . . . |   msg(lnk(e);tag(e);val(e))
| . . | . . . . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . |   ∃e':E.
| . . | . . . . |   isrcv(e')
| . . | . . . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . |                       ×ma-valtype(da; rcv(l; tg)
| . . | . . . . |                                    )))
```

```
| . . | . . . . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . |   ∈ {e':E
| . . | . . . . |      | isrcv(e')
| . . | . . . . |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . 35. ||map(LL;upto(||sends(l;e)||))|| = ||sends(l;e)||
| . . | . . . . 36. ||map(λe'.
| . . | . . . . |   <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . . . |   =
| . . | . . . . |   ||sends(l;e)||
| . . | . . . . 37. i1 : ℕ
| . . | . . . . 38. i1<||map(λe'.
| . . | . . . . |   <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . . . 39. v :
| . . | . . . . |   {e':E
| . . | . . . . |    | isrcv(e')
| . . | . . . . |      & lnk(e') = l
| . . | . . . . |        & sender(e') = e
| . . | . . . . |        & index(e') = i1 }
| . . | . . . . 40. LL(i1) = v
| . . | . . . . ⊢  <tag(v),val(v)> = 2of(sends(l;e)[i1])
| . . | . . . . | by Thin -1 THEN Analyze -1 THEN ExRepD
| . . | . . . . |
| . . | . . . . 1. i : Id
| . . | . . . . 2. k : Knd
| . . | . . . . 3. l : IdLnk
| . . | . . . . 4. ds : x:Id fp-> Type
| . . | . . . . 5. da : a:Knd fp-> Type
| . . | . . . . 6. f :
| . . | . . . . |  (tg:Id
| . . | . . . . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . 7. source(l) = i
| . . | . . . . 8. D' : Dsys
| . . | . . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . 10. w : World
| . . | . . . . 11. p : FairFifo
| . . | . . . . 12. PossibleWorld(D';w)
| . . | . . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . 14. ∀e:E.
| . . | . . . . |   loc(e) = i
| . . | . . . . |   ⇒
| . . | . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . 15. ∀e:E.
| . . | . . . . |   isrcv(e)
| . . | . . . . |   ⇒
| . . | . . . . |   lnk(e) = l
| . . | . . . . |   ⇒
| . . | . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . 17. ∀e:E.
| . . | . . . . |   loc(e) = i
| . . | . . . . |   ⇒
| . . | . . . . |   kind(e) = k
```

```
| . . | . . . . |   ⇒
| . . | . . . . |   sends(l;e)
| . . | . . . . |   =
| . . | . . . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . 19. ∀e:E.
| . . | . . . . |   loc(e) = i
| . . | . . . . |   ⇒
| . . | . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . 20. ∀e:E.
| . . | . . . . |   isrcv(e)
| . . | . . . . |   ⇒
| . . | . . . . |   lnk(e) = l
| . . | . . . . |   ⇒
| . . | . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . 21. e : E
| . . | . . . . 22. loc(e) = i
| . . | . . . . 23. kind(e) = k
| . . | . . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . 25. sends(l;e)
| . . | . . . . |   =
| . . | . . . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . 27. L :
| . . | . . . . |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . |   ∃e':E.
| . . | . . . . |   isrcv(e')
| . . | . . . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . 28. LL :
| . . | . . . . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . |                      | isrcv(e')
| . . | . . . . |                        & lnk(e') = l
| . . | . . . . |                          & sender(e') = e
| . . | . . . . |                          & index(e') = n }
| . . | . . . . 29. (λn.1of(L(n))) = LL
| . . | . . . . 30. ∀e:E.
| . . | . . . . |   isrcv(e)
| . . | . . . . |   ⇒
| . . | . . . . |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . |   =
| . . | . . . . |   msg(lnk(e);tag(e);val(e))
| . . | . . . . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . |   ∃e':E.
| . . | . . . . |   isrcv(e')
| . . | . . . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . |                        ×ma-valtype(da; rcv(l; tg)
| . . | . . . . |                                    )))
| . . | . . . . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . |   ∈ {e':E
| . . | . . . . |      | isrcv(e')
| . . | . . . . |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . 35. ||map(LL;upto(||sends(l;e)||))|| = ||sends(l;e)||
| . . | . . . . 36. ||map(λe'.
```

```
| . . | . . . . |   <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . . . |   =
| . . | . . . . |   ||sends(l;e)||
| . . | . . . . 37. i1 : ℕ
| . . | . . . . 38. i1<||map(λe'.
| . . | . . . . |   <tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))||
| . . | . . . . 39. v : E
| . . | . . . . 40. isrcv(v)
| . . | . . . . 41. lnk(v) = l
| . . | . . . . 42. sender(v) = e
| . . | . . . . 43. index(v) = i1
| . . | . . . . ⊢  <tag(v),val(v)> = 2of(sends(l;e)[i1])
| . . | . . . . | by Subst
| . . | . . . . |    (<tag(v),val(v)>
| . . | . . . . |     =
| . . | . . . . |     2of(msg(lnk(v);tag(v);val(v))))
| . . | . . . . |    0
| . . | . . . . |
| . . | . . . . |\
| . . | . . . . | .....equality .....
| . . | . . . . | 1. i : Id
| . . | . . . . | 2. k : Knd
| . . | . . . . | 3. l : IdLnk
| . . | . . . . | 4. ds : x:Id fp-> Type
| . . | . . . . | 5. da : a:Knd fp-> Type
| . . | . . . . | 6. f :
| . . | . . . . | |  (tg:Id
| . . | . . . . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . | 7. source(l) = i
| . . | . . . . | 8. D' : Dsys
| . . | . . . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | 10. w : World
| . . | . . . . | 11. p : FairFifo
| . . | . . . . | 12. PossibleWorld(D';w)
| . . | . . . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | 14. ∀e:E.
| . . | . . . . | |   loc(e) = i
| . . | . . . . | |   ⇒
| . . | . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | 15. ∀e:E.
| . . | . . . . | |   isrcv(e)
| . . | . . . . | |   ⇒
| . . | . . . . | |   lnk(e) = l
| . . | . . . . | |   ⇒
| . . | . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . | 17. ∀e:E.
| . . | . . . . | |   loc(e) = i
| . . | . . . . | |   ⇒
| . . | . . . . | |   kind(e) = k
| . . | . . . . | |   ⇒
| . . | . . . . | |   sends(l;e)
| . . | . . . . | |   =
```

```
| . . | . . . . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | 19. ∀e:E.
| . . | . . . . | |   loc(e) = i
| . . | . . . . | |   ⇒
| . . | . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | 20. ∀e:E.
| . . | . . . . | |   isrcv(e)
| . . | . . . . | |   ⇒
| . . | . . . . | |   lnk(e) = l
| . . | . . . . | |   ⇒
| . . | . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | 21. e : E
| . . | . . . . | 22. loc(e) = i
| . . | . . . . | 23. kind(e) = k
| . . | . . . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | 25. sends(l;e)
| . . | . . . . | |   =
| . . | . . . . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | 27. L :
| . . | . . . . | |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . | |   ∃e':E.
| . . | . . . . | |   isrcv(e')
| . . | . . . . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . | 28. LL :
| . . | . . . . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | |                       | isrcv(e')
| . . | . . . . | |                         & lnk(e') = l
| . . | . . . . | |                           & sender(e') = e
| . . | . . . . | |                           & index(e') = n }
| . . | . . . . | 29. (λn.1of(L(n))) = LL
| . . | . . . . | 30. ∀e:E.
| . . | . . . . | |   isrcv(e)
| . . | . . . . | |   ⇒
| . . | . . . . | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | |   =
| . . | . . . . | |   msg(lnk(e);tag(e);val(e))
| . . | . . . . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | |   ∃e':E.
| . . | . . . . | |   isrcv(e')
| . . | . . . . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | |                         ×ma-valtype(da; rcv(l; tg)
| . . | . . . . | |                                       )))
| . . | . . . . | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | |   ∈ {e':E
| . . | . . . . | |      | isrcv(e')
| . . | . . . . | |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | |   =
| . . | . . . . | |   ||sends(l;e)||
| . . | . . . . | 36. ||map(λe'.
| . . | . . . . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
```

```
| . . | . . . . . | |   =
| . . | . . . . . | |   ||sends(l;e)||
| . . | . . . . . | 37. i1 : ℕ
| . . | . . . . . | 38. i1<||map(λe'.
| . . | . . . . . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | 39. v : E
| . . | . . . . . | 40. isrcv(v)
| . . | . . . . . | 41. lnk(v) = l
| . . | . . . . . | 42. sender(v) = e
| . . | . . . . . | 43. index(v) = i1
| . . | . . . . . | ⊢  <tag(v),val(v)> = 2of(msg(lnk(v);tag(v);val(v)))
| . . | . . . . . | | by Unfold `msg` 0 THEN Reduce 0
| . . | . . . . . | |    THEN
| . . | . . . . . | |    Fold `member` 0
| . . | . . . . . | |    THEN
| . . | . . . . . | |    DoSubsume
| . . | . . . . . | |
| . . | . . . . . | .....subtype_rel .....
| . . | . . . . . | 1. i : Id
| . . | . . . . . | 2. k : Knd
| . . | . . . . . | 3. l : IdLnk
| . . | . . . . . | 4. ds : x:Id fp-> Type
| . . | . . . . . | 5. da : a:Knd fp-> Type
| . . | . . . . . | 6. f :
| . . | . . . . . | |  (tg:Id
| . . | . . . . . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . . | |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . . | 7. source(l) = i
| . . | . . . . . | 8. D' : Dsys
| . . | . . . . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . . | 10. w : World
| . . | . . . . . | 11. p : FairFifo
| . . | . . . . . | 12. PossibleWorld(D';w)
| . . | . . . . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | 14. ∀e:E.
| . . | . . . . . | |   loc(e) = i
| . . | . . . . . | |   ⇒
| . . | . . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | 15. ∀e:E.
| . . | . . . . . | |   isrcv(e)
| . . | . . . . . | |   ⇒
| . . | . . . . . | |   lnk(e) = l

| . . | . . . . . | |   ⇒
| . . | . . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | 17. ∀e:E.
| . . | . . . . . | |   loc(e) = i
| . . | . . . . . | |   ⇒
| . . | . . . . . | |   kind(e) = k
| . . | . . . . . | |   ⇒
| . . | . . . . . | |   sends(l;e)
| . . | . . . . . | |   =
| . . | . . . . . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
```

```
| . . | . . . . | 19. ∀e:E.
| . . | . . . . | |   loc(e) = i
| . . | . . . . | |   ⇒
| . . | . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | 20. ∀e:E.
| . . | . . . . | |   isrcv(e)
| . . | . . . . | |   ⇒
| . . | . . . . | |   lnk(e) = l
| . . | . . . . | |   ⇒
| . . | . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | 21. e : E
| . . | . . . . | 22. loc(e) = i
| . . | . . . . | 23. kind(e) = k
| . . | . . . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | 25. sends(l;e)
| . . | . . . . | |   =
| . . | . . . . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | 27. L :
| . . | . . . . | |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . | |   ∃e':E.
| . . | . . . . | |   isrcv(e')
| . . | . . . . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . | 28. LL :
| . . | . . . . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | |                       | isrcv(e')
| . . | . . . . | |                         & lnk(e') = l
| . . | . . . . | |                           & sender(e') = e
| . . | . . . . | |                           & index(e') = n }
| . . | . . . . | 29. (λn.1of(L(n))) = LL
| . . | . . . . | 30. ∀e:E.
| . . | . . . . | |   isrcv(e)
| . . | . . . . | |   ⇒
| . . | . . . . | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | |   =
| . . | . . . . | |   msg(lnk(e);tag(e);val(e))
| . . | . . . . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | |   ∃e':E.
| . . | . . . . | |   isrcv(e')
| . . | . . . . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | |                         ×ma-valtype(da; rcv(l; tg)
| . . | . . . . | |                                     )))
| . . | . . . . | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | |   ∈ {e':E
| . . | . . . . | |      | isrcv(e')
| . . | . . . . | |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | |   =
| . . | . . . . | |   ||sends(l;e)||
| . . | . . . . | 36. ||map(λe'.
| . . | . . . . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | |   =
| . . | . . . . | |   ||sends(l;e)||
```

```
| . . | . . . . | 37. i1 : ℕ
| . . | . . . . | 38. i1<||map(λe'.
| . . | . . . . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | 39. v : E
| . . | . . . . | 40. isrcv(v)
| . . | . . . . | 41. lnk(v) = l
| . . | . . . . | 42. sender(v) = e
| . . | . . . . | 43. index(v) = i1
| . . | . . . . | ⊢  valtype(v) ⊆r ma-valtype(da; rcv(l; tag(v)))
| . . | . . . . | | by Subst' (rcv(l; tag(v)) = kind(v)) 0
| . . | . . . . | |
| . . | . . . . | |\
| . . | . . . . | | 1. i : Id
| . . | . . . . | | 2. k : Knd
| . . | . . . . | | 3. l : IdLnk
| . . | . . . . | | 4. ds : x:Id fp-> Type
| . . | . . . . | | 5. da : a:Knd fp-> Type
| . . | . . . . | | 6. f :
| . . | . . . . | |    (tg:Id
| . . | . . . . | |     ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | |     (da(rcv(l; tg))?Void List)) List
| . . | . . . . | | 7. source(l) = i
| . . | . . . . | | 8. D' : Dsys
| . . | . . . . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | | 10. w : World
| . . | . . . . | | 11. p : FairFifo
| . . | . . . . | | 12. PossibleWorld(D';w)
| . . | . . . . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | 14. ∀e:E.
| . . | . . . . | |     loc(e) = i
| . . | . . . . | |     ⇒
| . . | . . . . | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | 15. ∀e:E.
| . . | . . . . | |     isrcv(e)
| . . | . . . . | |     ⇒
| . . | . . . . | |     lnk(e) = l
| . . | . . . . | |     ⇒
| . . | . . . . | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . | |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . | | 17. ∀e:E.
| . . | . . . . | |     loc(e) = i
| . . | . . . . | |     ⇒
| . . | . . . . | |     kind(e) = k
| . . | . . . . | |     ⇒
| . . | . . . . | |     sends(l;e)
| . . | . . . . | |     =
| . . | . . . . | |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | 19. ∀e:E.
| . . | . . . . | |     loc(e) = i
| . . | . . . . | |     ⇒
| . . | . . . . | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | 20. ∀e:E.
| . . | . . . . | |     isrcv(e)
```

```
| . . | . . . . | |     ⇒
| . . | . . . . | |     lnk(e) = l
| . . | . . . . | |     ⇒
| . . | . . . . | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | 21. e : E
| . . | . . . . | | 22. loc(e) = i
| . . | . . . . | | 23. kind(e) = k
| . . | . . . . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | 25. sends(l;e)
| . . | . . . . | |     =
| . . | . . . . | |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | 27. L :
| . . | . . . . | |     ∀n:ℕ||sends(l;e)||.
| . . | . . . . | |     ∃e':E.
| . . | . . . . | |     isrcv(e')
| . . | . . . . | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . | | 28. LL :
| . . | . . . . | |     n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | |                          | isrcv(e')
| . . | . . . . | |                            & lnk(e') = l
| . . | . . . . | |                              & sender(e') = e
| . . | . . . . | |                              & index(e') = n }
| . . | . . . . | | 29. (λn.1of(L(n))) = LL
| . . | . . . . | | 30. ∀e:E.
| . . | . . . . | |     isrcv(e)
| . . | . . . . | |     ⇒
| . . | . . . . | |     sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | |     =
| . . | . . . . | |     msg(lnk(e);tag(e);val(e))
| . . | . . . . | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | |     ∃e':E.
| . . | . . . . | |     isrcv(e')
| . . | . . . . | |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . | | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | |                            ×ma-valtype(da;
| . . | . . . . | |                                        rcv(l; tg)
| . . | . . . . | |                                        )))
| . . | . . . . | | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | |     ∈ {e':E
| . . | . . . . | |        | isrcv(e')
| . . | . . . . | |          & lnk(e') = l & sender(e') = e } List
| . . | . . . . | | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | |     =
| . . | . . . . | |     ||sends(l;e)||
| . . | . . . . | | 36. ||map(λe'.
| . . | . . . . | |     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | |     =
| . . | . . . . | |     ||sends(l;e)||
| . . | . . . . | | 37. i1 : ℕ
| . . | . . . . | | 38. i1<||map(λe'.
| . . | . . . . | |     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | 39. v : E
| . . | . . . . | | 40. isrcv(v)
```

```
| . . | . . . . | | 41. lnk(v) = l
| . . | . . . . | | 42. sender(v) = e
| . . | . . . . | | 43. index(v) = i1
| . . | . . . . | | ⊢  rcv(l; tag(v)) = kind(v)
| . . | . . . . | |   by Subst' (l = lnk(v)) 0
| . . | . . . . | |      THEN
| . . | . . . . | |      AssertBY isrcv(v) Trivial
| . . | . . . . | |      THEN
| . . | . . . . | |      MoveToConcl -1
| . . | . . . . | |      THEN
| . . | . . . . | |      Unfolds [`es-isrcv`;`es-tag`;`es-lnk`] 0
| . . | . . . . | |      THEN
| . . | . . . . | |      GenConclAtAddr [2;3]
| . . | . . . . | |      THEN
| . . | . . . . | |      Thin -1
| . . | . . . . | |      THEN
| . . | . . . . | |      Unfolds [`isrcv`;`lnk`;`tagof`] 0
| . . | . . . . | |      THEN
| . . | . . . . | |      Analyze -1
| . . | . . . . | |      THEN
| . . | . . . . | |      Analyze -1
| . . | . . . . | |      THEN
| . . | . . . . | |      Reduce 0
| . . | . . . . | |      THEN
| . . | . . . . | |      Fold `rcv` 0
| . . | . . . . | |      THEN
| . . | . . . . | |      Analyze
| . . | . . . . | |
| . . | . . . . | .\
| . . | . . . . | . 1. i : Id
| . . | . . . . | . 2. k : Knd
| . . | . . . . | . 3. l : IdLnk
| . . | . . . . | . 4. ds : x:Id fp-> Type
| . . | . . . . | . 5. da : a:Knd fp-> Type
| . . | . . . . | . 6. f :
| . . | . . . . | .    (tg:Id
| . . | . . . . | .     ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | .     (da(rcv(l; tg))?Void List)) List
| . . | . . . . | . 7. source(l) = i
| . . | . . . . | . 8. D' : Dsys
| . . | . . . . | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | . 10. w : World
| . . | . . . . | . 11. p : FairFifo
| . . | . . . . | . 12. PossibleWorld(D';w)
| . . | . . . . | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | . 14. ∀e:E.
| . . | . . . . | .     loc(e) = i
| . . | . . . . | .     ⇒
| . . | . . . . | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | . 15. ∀e:E.
| . . | . . . . | .     isrcv(e)
| . . | . . . . | .     ⇒
| . . | . . . . | .     lnk(e) = l
| . . | . . . . | .     ⇒
| . . | . . . . | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
```

```
| . . | . . . . . | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | .     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | . 17. ∀e:E.
| . . | . . . . . | .     loc(e) = i
| . . | . . . . . | .     ⇒
| . . | . . . . . | .     kind(e) = k
| . . | . . . . . | .     ⇒
| . . | . . . . . | .     sends(l;e)
| . . | . . . . . | .     =
| . . | . . . . . | .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | . 19. ∀e:E.
| . . | . . . . . | .     loc(e) = i
| . . | . . . . . | .     ⇒
| . . | . . . . . | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . 20. ∀e:E.
| . . | . . . . . | .     isrcv(e)
| . . | . . . . . | .     ⇒
| . . | . . . . . | .     lnk(e) = l
| . . | . . . . . | .     ⇒
| . . | . . . . . | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . 21. e : E
| . . | . . . . . | . 22. loc(e) = i
| . . | . . . . . | . 23. kind(e) = k
| . . | . . . . . | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | . 25. sends(l;e)
| . . | . . . . . | .     =
| . . | . . . . . | .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | . 27. L :
| . . | . . . . . | .     ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | .     ∃e':E.
| . . | . . . . . | .     isrcv(e')
| . . | . . . . . | .     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . | . 28. LL :
| . . | . . . . . | .     n:ℕ||sends(l;e)||→{e':E
| . . | . . . . . | .                         | isrcv(e')
| . . | . . . . . | .                           & lnk(e') = l
| . . | . . . . . | .                             & sender(e') = e
| . . | . . . . . | .                             & index(e') = n }
| . . | . . . . . | . 29. (λn.1of(L(n))) = LL
| . . | . . . . . | . 30. ∀e:E.
| . . | . . . . . | .     isrcv(e)
| . . | . . . . . | .     ⇒
| . . | . . . . . | .     sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . | .     =
| . . | . . . . . | .     msg(lnk(e);tag(e);val(e))
| . . | . . . . . | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . | .     ∃e':E.
| . . | . . . . . | .     isrcv(e')
| . . | . . . . . | .     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . | . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . . | .                           ×ma-valtype(da;
| . . | . . . . . | .                                       rcv(l; tg)
| . . | . . . . . | .                                       )))
```

```
| . . | . . . . . | . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . . | .     ∈ {e':E
| . . | . . . . . | .        | isrcv(e')
| . . | . . . . . | .          & lnk(e') = l & sender(e') = e } List
| . . | . . . . . | . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . . | . 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . . | .     =
| . . | . . . . . | .     ||sends(l;e)||
| . . | . . . . . | . 36. ||map(λe'.
| . . | . . . . . | .     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | .     =
| . . | . . . . . | .     ||sends(l;e)||
| . . | . . . . . | . 37. i1 : ℕ
| . . | . . . . . | . 38. i1<||map(λe'.
| . . | . . . . . | .     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | . 39. v : E
| . . | . . . . . | . 40. isrcv(v)
| . . | . . . . . | . 41. lnk(v) = l
| . . | . . . . . | . 42. sender(v) = e
| . . | . . . . . | . 43. index(v) = i1
| . . | . . . . . | . ⊢  valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | .   by BackThruSomeHyp
| . . | . . . . . |
| . . | . . . . . |\
| . . | . . . . . | 1. i : Id
| . . | . . . . . | 2. k : Knd
| . . | . . . . . | 3. l : IdLnk
| . . | . . . . . | 4. ds : x:Id fp-> Type
| . . | . . . . . | 5. da : a:Knd fp-> Type
| . . | . . . . . | 6. f :
| . . | . . . . . | |  (tg:Id
| . . | . . . . . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . . | |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . . | 7. source(l) = i
| . . | . . . . . | 8. D' : Dsys
| . . | . . . . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . . | 10. w : World
| . . | . . . . . | 11. p : FairFifo
| . . | . . . . . | 12. PossibleWorld(D';w)
| . . | . . . . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | 14. ∀e:E.
| . . | . . . . . | |   loc(e) = i
| . . | . . . . . | |   ⇒
| . . | . . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | 15. ∀e:E.
| . . | . . . . . | |   isrcv(e)
| . . | . . . . . | |   ⇒
| . . | . . . . . | |   lnk(e) = l
| . . | . . . . . | |   ⇒
| . . | . . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | 17. ∀e:E.
| . . | . . . . . | |   loc(e) = i
| . . | . . . . . | |   ⇒
```

```
| . . | . . . . | |   kind(e) = k
| . . | . . . . | |   ⇒
| . . | . . . . | |   sends(l;e)
| . . | . . . . | |   =
| . . | . . . . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | 19. ∀e:E.
| . . | . . . . | |   loc(e) = i
| . . | . . . . | |   ⇒
| . . | . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | 20. ∀e:E.
| . . | . . . . | |   isrcv(e)
| . . | . . . . | |   ⇒
| . . | . . . . | |   lnk(e) = l
| . . | . . . . | |   ⇒
| . . | . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | 21. e : E
| . . | . . . . | 22. loc(e) = i
| . . | . . . . | 23. kind(e) = k
| . . | . . . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | 25. sends(l;e)
| . . | . . . . | |   =
| . . | . . . . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | 27. L :
| . . | . . . . | |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . | |   ∃e':E.
| . . | . . . . | |   isrcv(e')
| . . | . . . . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . | 28. LL :
| . . | . . . . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | |                       | isrcv(e')
| . . | . . . . | |                         & lnk(e') = l
| . . | . . . . | |                           & sender(e') = e
| . . | . . . . | |                           & index(e') = n }
| . . | . . . . | 29. (λn.1of(L(n))) = LL
| . . | . . . . | 30. ∀e:E.
| . . | . . . . | |   isrcv(e)
| . . | . . . . | |   ⇒
| . . | . . . . | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | |   =
| . . | . . . . | |   msg(lnk(e);tag(e);val(e))
| . . | . . . . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | |   ∃e':E.
| . . | . . . . | |   isrcv(e')
| . . | . . . . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | |                         ×ma-valtype(da; rcv(l; tg)
| . . | . . . . | |                                      )))
| . . | . . . . | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | |   ∈ {e':E
| . . | . . . . | |      | isrcv(e')
| . . | . . . . | |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | 35. ||map(LL;upto(||sends(l;e)||))||
```

```
| . . | . . . . | |   =
| . . | . . . . | |   ||sends(l;e)||

| . . | . . . . | 36. ||map(λe'.
| . . | . . . . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | |   =
| . . | . . . . | |   ||sends(l;e)||
| . . | . . . . | 37. i1 : ℕ
| . . | . . . . | 38. i1<||map(λe'.
| . . | . . . . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | 39. v : E
| . . | . . . . | 40. isrcv(v)
| . . | . . . . | 41. lnk(v) = l
| . . | . . . . | 42. sender(v) = e
| . . | . . . . | 43. index(v) = i1
| . . | . . . . | ⊢  2of(msg(lnk(v);tag(v);val(v)))
| . . | . . . . | |  =
| . . | . . . . | |  2of(sends(l;e)[i1])
| . . | . . . . | | by AllHyps (λh.InstHyp [v] h THENA Complete Auto)
| . . | . . . . | |
| . . | . . . . | 1. i : Id
| . . | . . . . | 2. k : Knd
| . . | . . . . | 3. l : IdLnk
| . . | . . . . | 4. ds : x:Id fp-> Type
| . . | . . . . | 5. da : a:Knd fp-> Type
| . . | . . . . | 6. f :
| . . | . . . . | |  (tg:Id
| . . | . . . . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . | 7. source(l) = i
| . . | . . . . | 8. D' : Dsys
| . . | . . . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | 10. w : World
| . . | . . . . | 11. p : FairFifo
| . . | . . . . | 12. PossibleWorld(D';w)
| . . | . . . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | 14. ∀e:E.
| . . | . . . . | |   loc(e) = i
| . . | . . . . | |   ⇒
| . . | . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | 15. ∀e:E.
| . . | . . . . | |   isrcv(e)
| . . | . . . . | |   ⇒
| . . | . . . . | |   lnk(e) = l
| . . | . . . . | |   ⇒
| . . | . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . | 17. ∀e:E.
| . . | . . . . | |   loc(e) = i
| . . | . . . . | |   ⇒
| . . | . . . . | |   kind(e) = k
| . . | . . . . | |   ⇒
| . . | . . . . | |   sends(l;e)
| . . | . . . . | |   =
| . . | . . . . | |   tagged-messages(l;λz.(z when e);val(e);f)
```

```
| . . | . . . . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | 19. ∀e:E.
| . . | . . . . . | |   loc(e) = i
| . . | . . . . . | |   ⇒
| . . | . . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | 20. ∀e:E.
| . . | . . . . . | |   isrcv(e)
| . . | . . . . . | |   ⇒
| . . | . . . . . | |   lnk(e) = l
| . . | . . . . . | |   ⇒
| . . | . . . . . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | 21. e : E
| . . | . . . . . | 22. loc(e) = i
| . . | . . . . . | 23. kind(e) = k
| . . | . . . . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | 25. sends(l;e)
| . . | . . . . . | |   =
| . . | . . . . . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | 27. L :
| . . | . . . . . | |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | |   ∃e':E.
| . . | . . . . . | |   isrcv(e')
| . . | . . . . . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . | 28. LL :
| . . | . . . . . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . . | |                      | isrcv(e')
| . . | . . . . . | |                        & lnk(e') = l
| . . | . . . . . | |                          & sender(e') = e
| . . | . . . . . | |                          & index(e') = n }
| . . | . . . . . | 29. (λn.1of(L(n))) = LL
| . . | . . . . . | 30. ∀e:E.
| . . | . . . . . | |   isrcv(e)
| . . | . . . . . | |   ⇒
| . . | . . . . . | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . | |   =
| . . | . . . . . | |   msg(lnk(e);tag(e);val(e))
| . . | . . . . . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . | |   ∃e':E.
| . . | . . . . . | |   isrcv(e')
| . . | . . . . . | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . . | |                         ×ma-valtype(da; rcv(l; tg)
| . . | . . . . . | |                                       )))
| . . | . . . . . | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . . | |   ∈ {e':E
| . . | . . . . . | |      | isrcv(e')

| . . | . . . . . | |       & lnk(e') = l & sender(e') = e } List
| . . | . . . . . | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . . | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . . | |   =
| . . | . . . . . | |   ||sends(l;e)||
| . . | . . . . . | 36. ||map(λe'.
| . . | . . . . . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | |   =
```

```
| . . | . . . . | |   ||sends(l;e)||
| . . | . . . . | 37. i1 : ℕ
| . . | . . . . | 38. i1<||map(λe'.
| . . | . . . . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | 39. v : E
| . . | . . . . | 40. isrcv(v)
| . . | . . . . | 41. lnk(v) = l
| . . | . . . . | 42. sender(v) = e
| . . | . . . . | 43. index(v) = i1
| . . | . . . . | 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | |   ∃e':E.
| . . | . . . . | |   isrcv(e')
| . . | . . . . | |   & lnk(e') = l & sender(e') = v & index(e') = n
| . . | . . . . | 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | |   =
| . . | . . . . | |   msg(lnk(v);tag(v);val(v))
| . . | . . . . | 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | ⊢  2of(msg(lnk(v);tag(v);val(v)))
| . . | . . . . | |  =
| . . | . . . . | |  2of(sends(l;e)[i1])
| . . | . . . . | | by Assert
| . . | . . . . | |    (msg(lnk(v);tag(v);val(v)) = sends(l;e)[i1])
| . . | . . . . | |
| . . | . . . . | |\
| . . | . . . . | | .....assertion .....
| . . | . . . . | | 1. i : Id
| . . | . . . . | | 2. k : Knd
| . . | . . . . | | 3. l : IdLnk
| . . | . . . . | | 4. ds : x:Id fp-> Type
| . . | . . . . | | 5. da : a:Knd fp-> Type
| . . | . . . . | | 6. f :
| . . | . . . . | | |  (tg:Id
| . . | . . . . | | |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | | |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . | | 7. source(l) = i
| . . | . . . . | | 8. D' : Dsys
| . . | . . . . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | | 10. w : World
| . . | . . . . | | 11. p : FairFifo
| . . | . . . . | | 12. PossibleWorld(D';w)
| . . | . . . . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | 14. ∀e:E.
| . . | . . . . | | |   loc(e) = i
| . . | . . . . | | |   ⇒
| . . | . . . . | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | 15. ∀e:E.
| . . | . . . . | | |   isrcv(e)
| . . | . . . . | | |   ⇒
| . . | . . . . | | |   lnk(e) = l
| . . | . . . . | | |   ⇒
| . . | . . . . | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . | | 17. ∀e:E.
```

```
| . . | . . . . | | |   loc(e) = i
| . . | . . . . | | |   ⇒
| . . | . . . . | | |   kind(e) = k
| . . | . . . . | | |   ⇒
| . . | . . . . | | |   sends(l;e)
| . . | . . . . | | |   =
| . . | . . . . | | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | 19. ∀e:E.
| . . | . . . . | | |   loc(e) = i
| . . | . . . . | | |   ⇒
| . . | . . . . | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | 20. ∀e:E.
| . . | . . . . | | |   isrcv(e)
| . . | . . . . | | |   ⇒
| . . | . . . . | | |   lnk(e) = l
| . . | . . . . | | |   ⇒
| . . | . . . . | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | 21. e : E
| . . | . . . . | | 22. loc(e) = i
| . . | . . . . | | 23. kind(e) = k
| . . | . . . . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | 25. sends(l;e)
| . . | . . . . | | |   =
| . . | . . . . | | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | 27. L :
| . . | . . . . | | |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . | | |   ∃e':E.
| . . | . . . . | | |   isrcv(e')
| . . | . . . . | | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . | | 28. LL :
| . . | . . . . | | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | | |                        | isrcv(e')
| . . | . . . . | | |                          & lnk(e') = l
| . . | . . . . | | |                            & sender(e') = e
| . . | . . . . | | |                            & index(e') = n }

| . . | . . . . | | 29. (λn.1of(L(n))) = LL
| . . | . . . . | | 30. ∀e:E.
| . . | . . . . | | |   isrcv(e)
| . . | . . . . | | |   ⇒
| . . | . . . . | | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | | |   =
| . . | . . . . | | |   msg(lnk(e);tag(e);val(e))
| . . | . . . . | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | | |   ∃e':E.
| . . | . . . . | | |   isrcv(e')
| . . | . . . . | | |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . | | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | | |                          ×ma-valtype(da;
| . . | . . . . | | |                                      rcv(l; tg)
| . . | . . . . | | |                                      )))
| . . | . . . . | | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | | |   ∈ {e':E
| . . | . . . . | | |      | isrcv(e')
```

```
| . . | . . . . | | |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . | | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | | |   =
| . . | . . . . | | |   ||sends(l;e)||
| . . | . . . . | | 36. ||map(λe'.
| . . | . . . . | | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | |   =
| . . | . . . . | | |   ||sends(l;e)||
| . . | . . . . | | 37. i1 : ℕ
| . . | . . . . | | 38. i1<||map(λe'.
| . . | . . . . | | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | 39. v : E
| . . | . . . . | | 40. isrcv(v)
| . . | . . . . | | 41. lnk(v) = l
| . . | . . . . | | 42. sender(v) = e
| . . | . . . . | | 43. index(v) = i1
| . . | . . . . | | 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | | |   ∃e':E.
| . . | . . . . | | |   isrcv(e')
| . . | . . . . | | |   & lnk(e') = l & sender(e') = v & index(e') = n
| . . | . . . . | | 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | |   =
| . . | . . . . | | |   msg(lnk(v);tag(v);val(v))
| . . | . . . . | | 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | ⊢  msg(lnk(v);tag(v);val(v)) = sends(l;e)[i1]
| . . | . . . . | | | by StrongRevHypSubst -3 0
| . . | . . . . | | |
| . . | . . . . | | |\
| . . | . . . . | | | 1. i : Id
| . . | . . . . | | | 2. k : Knd
| . . | . . . . | | | 3. l : IdLnk
| . . | . . . . | | | 4. ds : x:Id fp-> Type
| . . | . . . . | | | 5. da : a:Knd fp-> Type
| . . | . . . . | | | 6. f :
| . . | . . . . | | | |  (tg:Id
| . . | . . . . | | | |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | | | |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . | | | 7. source(l) = i
| . . | . . . . | | | 8. D' : Dsys
| . . | . . . . | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | | | 10. w : World
| . . | . . . . | | | 11. p : FairFifo
| . . | . . . . | | | 12. PossibleWorld(D';w)
| . . | . . . . | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | | 14. ∀e:E.
| . . | . . . . | | | |   loc(e) = i
| . . | . . . . | | | |   ⇒
| . . | . . . . | | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | 15. ∀e:E.
| . . | . . . . | | | |   isrcv(e)
| . . | . . . . | | | |   ⇒
| . . | . . . . | | | |   lnk(e) = l
| . . | . . . . | | | |   ⇒
```

```
| . . | . . . . | | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . | | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . | | | 17. ∀e:E.
| . . | . . . . | | | |   loc(e) = i
| . . | . . . . | | | |   ⇒
| . . | . . . . | | | |   kind(e) = k
| . . | . . . . | | | |   ⇒
| . . | . . . . | | | |   sends(l;e)
| . . | . . . . | | | |   =
| . . | . . . . | | | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | | 19. ∀e:E.
| . . | . . . . | | | |   loc(e) = i
| . . | . . . . | | | |   ⇒
| . . | . . . . | | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | 20. ∀e:E.
| . . | . . . . | | | |   isrcv(e)
| . . | . . . . | | | |   ⇒
| . . | . . . . | | | |   lnk(e) = l
| . . | . . . . | | | |   ⇒
| . . | . . . . | | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | 21. e : E
| . . | . . . . | | | 22. loc(e) = i
| . . | . . . . | | | 23. kind(e) = k
| . . | . . . . | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | | 25. sends(l;e)
| . . | . . . . | | | |   =
| . . | . . . . | | | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | | 27. L :
| . . | . . . . | | | |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . | | | |   ∃e':E.
| . . | . . . . | | | |   isrcv(e')
| . . | . . . . | | | |   & lnk(e') = l
| . . | . . . . | | | |     & sender(e') = e
| . . | . . . . | | | |     & index(e') = n
| . . | . . . . | | | 28. LL :
| . . | . . . . | | | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | | | |                       | isrcv(e')
| . . | . . . . | | | |                         & lnk(e') = l
| . . | . . . . | | | |                           & sender(e') = e
| . . | . . . . | | | |                           & index(e') = n }
| . . | . . . . | | | 29. (λn.1of(L(n))) = LL
| . . | . . . . | | | 30. ∀e:E.
| . . | . . . . | | | |   isrcv(e)
| . . | . . . . | | | |   ⇒
| . . | . . . . | | | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | | | |   =
| . . | . . . . | | | |   msg(lnk(e);tag(e);val(e))
| . . | . . . . | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | | | |   ∃e':E.
| . . | . . . . | | | |   isrcv(e')
| . . | . . . . | | | |   & lnk(e') = l
| . . | . . . . | | | |     & sender(e') = e
```

```
| . . | . . . . | | | |     & index(e’) = n
| . . | . . . . | | | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | | | |                         ×ma-valtype(da;
| . . | . . . . | | | |                                     rcv(l; tg)
| . . | . . . . | | | |                                     )))
| . . | . . . . | | | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | | | |   ∈ {e’:E
| . . | . . . . | | | |      | isrcv(e’)
| . . | . . . . | | | |        & lnk(e’) = l & sender(e’) = e } List
| . . | . . . . | | | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | | | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | | | |   =
| . . | . . . . | | | |   ||sends(l;e)||
| . . | . . . . | | | 36. ||map(λe’.
| . . | . . . . | | | |   <tag(e’),val(e’)>;map(LL;upto(||...||)))||
| . . | . . . . | | | |   =
| . . | . . . . | | | |   ||sends(l;e)||
| . . | . . . . | | | 37. i1 : ℕ
| . . | . . . . | | | 38. i1<||map(λe’.
| . . | . . . . | | | |   <tag(e’),val(e’)>;map(LL;upto(||...||)))||
| . . | . . . . | | | 39. v : E
| . . | . . . . | | | 40. isrcv(v)
| . . | . . . . | | | 41. lnk(v) = l
| . . | . . . . | | | 42. sender(v) = e
| . . | . . . . | | | 43. index(v) = i1
| . . | . . . . | | | 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | | | |   ∃e’:E.
| . . | . . . . | | | |   isrcv(e’)
| . . | . . . . | | | |   & lnk(e’) = l
| . . | . . . . | | | |     & sender(e’) = v
| . . | . . . . | | | |     & index(e’) = n
| . . | . . . . | | | 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | | |   =
| . . | . . . . | | | |   msg(lnk(v);tag(v);val(v))
| . . | . . . . | | | 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | | 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | | ⊢  sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | | |  =
| . . | . . . . | | | |  sends(l;e)[i1]  by Analyze
| . . | . . . . | | | |
| . . | . . . . | | | |\
| . . | . . . . | | | | .....subterm .....
| . . | . . . . | | | | 1. i : Id
| . . | . . . . | | | | 2. k : Knd
| . . | . . . . | | | | 3. l : IdLnk
| . . | . . . . | | | | 4. ds : x:Id fp-> Type
| . . | . . . . | | | | 5. da : a:Knd fp-> Type
| . . | . . . . | | | | 6. f :
| . . | . . . . | | | | |  (tg:Id
| . . | . . . . | | | | |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | | | | |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . | | | | 7. source(l) = i
| . . | . . . . | | | | 8. D’ : Dsys
| . . | . . . . | | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D’
| . . | . . . . | | | | 10. w : World
```

```
| . . | . . . . | | | | 11. p : FairFifo
| . . | . . . . | | | | 12. PossibleWorld(D';w)
| . . | . . . . | | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | | | 14. ∀e:E.
| . . | . . . . | | | | |   loc(e) = i
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | | 15. ∀e:E.
| . . | . . . . | | | | |   isrcv(e)
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   lnk(e) = l
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . | | | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . | | | | 17. ∀e:E.
| . . | . . . . | | | | |   loc(e) = i
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   kind(e) = k
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   sends(l;e)
| . . | . . . . | | | | |   =
| . . | . . . . | | | | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | | | 19. ∀e:E.
| . . | . . . . | | | | |   loc(e) = i
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | | 20. ∀e:E.
| . . | . . . . | | | | |   isrcv(e)
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   lnk(e) = l
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | | 21. e : E
| . . | . . . . | | | | 22. loc(e) = i
| . . | . . . . | | | | 23. kind(e) = k
| . . | . . . . | | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | | | 25. sends(l;e)
| . . | . . . . | | | | |   =
| . . | . . . . | | | | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | | | 27. L :
| . . | . . . . | | | | |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . | | | | |   ∃e':E.
| . . | . . . . | | | | |   isrcv(e')
| . . | . . . . | | | | |   & lnk(e') = l
| . . | . . . . | | | | |     & sender(e') = e
| . . | . . . . | | | | |     & index(e') = n
| . . | . . . . | | | | 28. LL :
| . . | . . . . | | | | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | | | | |                     | isrcv(e')
| . . | . . . . | | | | |                       & lnk(e') = l
| . . | . . . . | | | | |                         & sender(e') = e
| . . | . . . . | | | | |                         & index(e') = n }
```

```
| . . | . . . . | | | | 29. (λn.1of(L(n))) = LL
| . . | . . . . | | | | 30. ∀e:E.
| . . | . . . . | | | | |   isrcv(e)
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | | | | |   =
| . . | . . . . | | | | |   msg(lnk(e);tag(e);val(e))
| . . | . . . . | | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | | | | |   ∃e':E.
| . . | . . . . | | | | |   isrcv(e')
| . . | . . . . | | | | |   & lnk(e') = l
| . . | . . . . | | | | |     & sender(e') = e
| . . | . . . . | | | | |     & index(e') = n
| . . | . . . . | | | | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | | | | |                       ×ma-valtype(da;
| . . | . . . . | | | | |                                   rcv(l;
| . . | . . . . | | | | |                                       tg)
| . . | . . . . | | | | |                                   )))
| . . | . . . . | | | | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | | | | |   ∈ {e':E
| . . | . . . . | | | | |      | isrcv(e')
| . . | . . . . | | | | |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . | | | | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | | | | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | | | | |   =
| . . | . . . . | | | | |   ||sends(l;e)||
| . . | . . . . | | | | 36. ||map(λe'.
| . . | . . . . | | | | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | | | |   =
| . . | . . . . | | | | |   ||sends(l;e)||
| . . | . . . . | | | | 37. i1 : ℕ
| . . | . . . . | | | | 38. i1<||map(λe'.
| . . | . . . . | | | | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | | | 39. v : E
| . . | . . . . | | | | 40. isrcv(v)
| . . | . . . . | | | | 41. lnk(v) = l
| . . | . . . . | | | | 42. sender(v) = e
| . . | . . . . | | | | 43. index(v) = i1
| . . | . . . . | | | | 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | | | | |   ∃e':E.
| . . | . . . . | | | | |   isrcv(e')
| . . | . . . . | | | | |   & lnk(e') = l
| . . | . . . . | | | | |     & sender(e') = v
| . . | . . . . | | | | |     & index(e') = n
| . . | . . . . | | | | 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | | | |   =
| . . | . . . . | | | | |   msg(lnk(v);tag(v);val(v))
| . . | . . . . | | | | 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | | | 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | | | ⊢  sends(lnk(v);sender(v)) = sends(l;e)
| . . | . . . . | | | | | by Analyze
| . . | . . . . | | | | |
| . . | . . . . | | | | .....unproved Inclusion subgoal .....
| . . | . . . . | | | | 1. i : Id
| . . | . . . . | | | | 2. k : Knd
```

```
| . . | . . . . | | | | 3. l : IdLnk
| . . | . . . . | | | | 4. ds : x:Id fp-> Type
| . . | . . . . | | | | 5. da : a:Knd fp-> Type
| . . | . . . . | | | | 6. f :
| . . | . . . . | | | | |  (tg:Id
| . . | . . . . | | | | |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | | | | |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . | | | | 7. source(l) = i
| . . | . . . . | | | | 8. D' : Dsys
| . . | . . . . | | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | | | | 10. w : World
| . . | . . . . | | | | 11. p : FairFifo
| . . | . . . . | | | | 12. PossibleWorld(D';w)
| . . | . . . . | | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | | | 14. ∀e:E.
| . . | . . . . | | | | |   loc(e) = i
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | | 15. ∀e:E.
| . . | . . . . | | | | |   isrcv(e)
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   lnk(e) = l
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . | | | | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . | | | | 17. ∀e:E.
| . . | . . . . | | | | |   loc(e) = i
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   kind(e) = k
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   sends(l;e)
| . . | . . . . | | | | |   =
| . . | . . . . | | | | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | | | 19. ∀e:E.
| . . | . . . . | | | | |   loc(e) = i
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | | 20. ∀e:E.
| . . | . . . . | | | | |   isrcv(e)
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   lnk(e) = l
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | | 21. e : E
| . . | . . . . | | | | 22. loc(e) = i
| . . | . . . . | | | | 23. kind(e) = k
| . . | . . . . | | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | | | 25. sends(l;e)
| . . | . . . . | | | | |   =
| . . | . . . . | | | | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | | | 27. L :
| . . | . . . . | | | | |   ∀n:ℕ||sends(l;e)||.
```

```
| . . | . . . . | | | | |   ∃e':E.
| . . | . . . . | | | | |   isrcv(e')
| . . | . . . . | | | | |   & lnk(e') = l
| . . | . . . . | | | | |     & sender(e') = e
| . . | . . . . | | | | |     & index(e') = n
| . . | . . . . | | | | 28. LL :
| . . | . . . . | | | | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | | | | |                       | isrcv(e')
| . . | . . . . | | | | |                         & lnk(e') = l
| . . | . . . . | | | | |                           & sender(e') = e
| . . | . . . . | | | | |                           & index(e') = n }
| . . | . . . . | | | | 29. (λn.1of(L(n))) = LL
| . . | . . . . | | | | 30. ∀e:E.
| . . | . . . . | | | | |   isrcv(e)
| . . | . . . . | | | | |   ⇒
| . . | . . . . | | | | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | | | | |   =
| . . | . . . . | | | | |   msg(lnk(e);tag(e);val(e))
| . . | . . . . | | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | | | | |   ∃e':E.
| . . | . . . . | | | | |   isrcv(e')
| . . | . . . . | | | | |   & lnk(e') = l
| . . | . . . . | | | | |     & sender(e') = e
| . . | . . . . | | | | |     & index(e') = n
| . . | . . . . | | | | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | | | | |                         ×ma-valtype(da;
| . . | . . . . | | | | |                                     rcv(l;
| . . | . . . . | | | | |                                         tg)
| . . | . . . . | | | | |                                     )))
| . . | . . . . | | | | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | | | | |   ∈ {e':E
| . . | . . . . | | | | |      | isrcv(e')
| . . | . . . . | | | | |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . | | | | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | | | | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | | | | |   =
| . . | . . . . | | | | |   ||sends(l;e)||
| . . | . . . . | | | | 36. ||map(λe'.
| . . | . . . . | | | | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | | | |   =
| . . | . . . . | | | | |   ||sends(l;e)||
| . . | . . . . | | | | 37. i1 : ℕ
| . . | . . . . | | | | 38. i1<||map(λe'.
| . . | . . . . | | | | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | | | 39. v : E
| . . | . . . . | | | | 40. isrcv(v)
| . . | . . . . | | | | 41. lnk(v) = l
| . . | . . . . | | | | 42. sender(v) = e
| . . | . . . . | | | | 43. index(v) = i1
| . . | . . . . | | | | 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | | | | |   ∃e':E.
| . . | . . . . | | | | |   isrcv(e')
| . . | . . . . | | | | |   & lnk(e') = l
| . . | . . . . | | | | |     & sender(e') = v
| . . | . . . . | | | | |     & index(e') = n
```

```
| . . | . . . . | | | | 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | | | |   =
| . . | . . . . | | | | |   msg(lnk(v);tag(v);val(v))
| . . | . . . . | | | | 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | | | 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | | | 48. sends(lnk(v);sender(v)) = sends(l;e)
| . . | . . . . | | | | 49. (Msg on lnk(v)) List
| . . | . . . . | | | | 50. (Msg on lnk(v)) List
| . . | . . . . | | | | 51. u : (Msg on lnk(v))
| . . | . . . . | | | | ⊢  u = u
| . . | . . . . | | | | | by Fold `member` 0 THEN DoSubsume
| . . | . . . . | | | | |    THEN
| . . | . . . . | | | | |    Subst' (lnk(v) = l) 0
| . . | . . . . | | | | |
| . . | . . . . | | | | 1. i : Id
| . . | . . . . | | | | 2. k : Knd
| . . | . . . . | | | | 3. l : IdLnk
| . . | . . . . | | | | 4. ds : x:Id fp-> Type
| . . | . . . . | | | | 5. da : a:Knd fp-> Type
| . . | . . . . | | | | 6. f :
| . . | . . . . | | | |    (tg:Id
| . . | . . . . | | | |     ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | | | |     (da(rcv(l; tg))?Void List)) List
| . . | . . . . | | | | 7. source(l) = i
| . . | . . . . | | | | 8. D' : Dsys
| . . | . . . . | | | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | | | | 10. w : World
| . . | . . . . | | | | 11. p : FairFifo
| . . | . . . . | | | | 12. PossibleWorld(D';w)
| . . | . . . . | | | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | | | 14. ∀e:E.
| . . | . . . . | | | |     loc(e) = i
| . . | . . . . | | | |     ⇒
| . . | . . . . | | | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | | 15. ∀e:E.
| . . | . . . . | | | |     isrcv(e)
| . . | . . . . | | | |     ⇒
| . . | . . . . | | | |     lnk(e) = l
| . . | . . . . | | | |     ⇒
| . . | . . . . | | | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . | | | |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . | | | | 17. ∀e:E.
| . . | . . . . | | | |     loc(e) = i
| . . | . . . . | | | |     ⇒
| . . | . . . . | | | |     kind(e) = k
| . . | . . . . | | | |     ⇒
| . . | . . . . | | | |     sends(l;e)
| . . | . . . . | | | |     =
| . . | . . . . | | | |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | | | 19. ∀e:E.
| . . | . . . . | | | |     loc(e) = i
| . . | . . . . | | | |     ⇒
| . . | . . . . | | | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
```

```
| . . | . . . . . | | | | 20. ∀e:E.
| . . | . . . . . | | | |     isrcv(e)
| . . | . . . . . | | | |     ⇒
| . . | . . . . . | | | |     lnk(e) = l
| . . | . . . . . | | | |     ⇒
| . . | . . . . . | | | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | | | 21. e : E
| . . | . . . . . | | | | 22. loc(e) = i
| . . | . . . . . | | | | 23. kind(e) = k
| . . | . . . . . | | | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | | | 25. sends(l;e)
| . . | . . . . . | | | |     =
| . . | . . . . . | | | |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | | | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | | | 27. L :
| . . | . . . . . | | | |     ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | | | |     ∃e':E.
| . . | . . . . . | | | |     isrcv(e')
| . . | . . . . . | | | |     & lnk(e') = l
| . . | . . . . . | | | |       & sender(e') = e
| . . | . . . . . | | | |       & index(e') = n
| . . | . . . . . | | | | 28. LL :
| . . | . . . . . | | | |     n:ℕ||sends(l;e)||→{e':E
| . . | . . . . . | | | |                          | isrcv(e')
| . . | . . . . . | | | |                            & lnk(e') = l
| . . | . . . . . | | | |                              & sender(e') = e
| . . | . . . . . | | | |                              & index(e') = n }
| . . | . . . . . | | | | 29. (λn.1of(L(n))) = LL
| . . | . . . . . | | | | 30. ∀e:E.
| . . | . . . . . | | | |     isrcv(e)
| . . | . . . . . | | | |     ⇒
| . . | . . . . . | | | |     sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . | | | |     =
| . . | . . . . . | | | |     msg(lnk(e);tag(e);val(e))
| . . | . . . . . | | | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . | | | |     ∃e':E.
| . . | . . . . . | | | |     isrcv(e')
| . . | . . . . . | | | |     & lnk(e') = l
| . . | . . . . . | | | |       & sender(e') = e
| . . | . . . . . | | | |       & index(e') = n
| . . | . . . . . | | | | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . . | | | |                             ×ma-valtype(da;

| . . | . . . . . | | | |                                                rcv(l;
| . . | . . . . . | | | |                                                    tg)
| . . | . . . . . | | | |                                                )))
| . . | . . . . . | | | | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . . | | | |     ∈ {e':E
| . . | . . . . . | | | |        | isrcv(e')
| . . | . . . . . | | | |          & lnk(e') = l & sender(e') = e } List
| . . | . . . . . | | | | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . . | | | | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . . | | | |     =
| . . | . . . . . | | | |     ||sends(l;e)||
| . . | . . . . . | | | | 36. ||map(λe'.
| . . | . . . . . | | | |     <tag(e'),val(e')>;map(LL;upto(||...||)))||
```

```
| . . | . . . . | | | |     =
| . . | . . . . | | | |     ||sends(l;e)||
| . . | . . . . | | | | 37. i1 : ℕ
| . . | . . . . | | | | 38. i1<||map(λe'.
| . . | . . . . | | | |     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | | | 39. v : E
| . . | . . . . | | | | 40. isrcv(v)
| . . | . . . . | | | | 41. lnk(v) = l
| . . | . . . . | | | | 42. sender(v) = e
| . . | . . . . | | | | 43. index(v) = i1
| . . | . . . . | | | | 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | | | |     ∃e':E.
| . . | . . . . | | | |     isrcv(e')
| . . | . . . . | | | |     & lnk(e') = l
| . . | . . . . | | | |       & sender(e') = v
| . . | . . . . | | | |       & index(e') = n
| . . | . . . . | | | | 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | | |     =
| . . | . . . . | | | |     msg(lnk(v);tag(v);val(v))
| . . | . . . . | | | | 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | | | 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | | | 48. sends(lnk(v);sender(v)) = sends(l;e)
| . . | . . . . | | | | 49. (Msg on lnk(v)) List
| . . | . . . . | | | | 50. (Msg on lnk(v)) List
| . . | . . . . | | | | 51. (Msg on lnk(v))
| . . | . . . . | | | | ⊢  (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | | | |                          ×ma-valtype(da;
| . . | . . . . | | | |                                       rcv(l; tg
| . . | . . . . | | | |                                           ))))
| . . | . . . . | | | |   by Auto
| . . | . . . . | | | |
| . . | . . . . | | | .\
| . . | . . . . | | | . 1. i : Id
| . . | . . . . | | | . 2. k : Knd
| . . | . . . . | | | . 3. l : IdLnk
| . . | . . . . | | | . 4. ds : x:Id fp-> Type
| . . | . . . . | | | . 5. da : a:Knd fp-> Type
| . . | . . . . | | | . 6. f :
| . . | . . . . | | | .    (tg:Id
| . . | . . . . | | | .     ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | | | .     (da(rcv(l; tg))?Void List)) List
| . . | . . . . | | | . 7. source(l) = i
| . . | . . . . | | | . 8. D' : Dsys
| . . | . . . . | | | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | | | . 10. w : World
| . . | . . . . | | | . 11. p : FairFifo
| . . | . . . . | | | . 12. PossibleWorld(D';w)
| . . | . . . . | | | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | | . 14. ∀e:E.
| . . | . . . . | | | .     loc(e) = i
| . . | . . . . | | | .     ⇒
| . . | . . . . | | | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | . 15. ∀e:E.
| . . | . . . . | | | .     isrcv(e)
| . . | . . . . | | | .     ⇒
```

```
| . . | . . . . | | | .     lnk(e) = l
| . . | . . . . | | | .     ⇒
| . . | . . . . | | | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . | | | .     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . | | | . 17. ∀e:E.
| . . | . . . . | | | .     loc(e) = i
| . . | . . . . | | | .     ⇒
| . . | . . . . | | | .     kind(e) = k
| . . | . . . . | | | .     ⇒
| . . | . . . . | | | .     sends(l;e)
| . . | . . . . | | | .     =
| . . | . . . . | | | .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | | . 19. ∀e:E.
| . . | . . . . | | | .     loc(e) = i
| . . | . . . . | | | .     ⇒
| . . | . . . . | | | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | . 20. ∀e:E.
| . . | . . . . | | | .     isrcv(e)
| . . | . . . . | | | .     ⇒
| . . | . . . . | | | .     lnk(e) = l
| . . | . . . . | | | .     ⇒
| . . | . . . . | | | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | | . 21. e : E
| . . | . . . . | | | . 22. loc(e) = i
| . . | . . . . | | | . 23. kind(e) = k
| . . | . . . . | | | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | | . 25. sends(l;e)
| . . | . . . . | | | .     =
| . . | . . . . | | | .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | | . 27. L :
| . . | . . . . | | | .     ∀n:ℕ||sends(l;e)||.
| . . | . . . . | | | .     ∃e':E.
| . . | . . . . | | | .     isrcv(e')
| . . | . . . . | | | .     & lnk(e') = l
| . . | . . . . | | | .       & sender(e') = e
| . . | . . . . | | | .       & index(e') = n
| . . | . . . . | | | . 28. LL :
| . . | . . . . | | | .     n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | | | .                         | isrcv(e')
| . . | . . . . | | | .                           & lnk(e') = l
| . . | . . . . | | | .                             & sender(e') = e
| . . | . . . . | | | .                             & index(e') = n }
| . . | . . . . | | | . 29. (λn.1of(L(n))) = LL
| . . | . . . . | | | . 30. ∀e:E.
| . . | . . . . | | | .     isrcv(e)
| . . | . . . . | | | .     ⇒
| . . | . . . . | | | .     sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | | | .     =
| . . | . . . . | | | .     msg(lnk(e);tag(e);val(e))
| . . | . . . . | | | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | | | .     ∃e':E.
| . . | . . . . | | | .     isrcv(e')
```

```
| . . | . . . . | | | .     & lnk(e') = l
| . . | . . . . | | | .       & sender(e') = e
| . . | . . . . | | | .       & index(e') = n
| . . | . . . . | | | . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | | | .                          ×ma-valtype(da;
| . . | . . . . | | | .                                       rcv(l;
| . . | . . . . | | | .                                           tg)
| . . | . . . . | | | .                                       )))
| . . | . . . . | | | . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | | | .     ∈ {e':E
| . . | . . . . | | | .        | isrcv(e')
| . . | . . . . | | | .          & lnk(e') = l & sender(e') = e } List
| . . | . . . . | | | . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | | | . 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | | | .     =
| . . | . . . . | | | .     ||sends(l;e)||
| . . | . . . . | | | . 36. ||map(λe'.
| . . | . . . . | | | .     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | | .     =
| . . | . . . . | | | .     ||sends(l;e)||
| . . | . . . . | | | . 37. i1 : ℕ
| . . | . . . . | | | . 38. i1<||map(λe'.
| . . | . . . . | | | .     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | | . 39. v : E
| . . | . . . . | | | . 40. isrcv(v)
| . . | . . . . | | | . 41. lnk(v) = l
| . . | . . . . | | | . 42. sender(v) = e
| . . | . . . . | | | . 43. index(v) = i1
| . . | . . . . | | | . 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | | | .     ∃e':E.
| . . | . . . . | | | .     isrcv(e')
| . . | . . . . | | | .     & lnk(e') = l
| . . | . . . . | | | .       & sender(e') = v
| . . | . . . . | | | .       & index(e') = n
| . . | . . . . | | | . 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | | .     =
| . . | . . . . | | | .     msg(lnk(v);tag(v);val(v))
| . . | . . . . | | | . 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | | . 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | | . ⊢  index(v)<||sends(lnk(v);sender(v))||
| . . | . . . . | | | .   by HypSubst' -7 0 THEN HypSubst -6 0
| . . | . . . . | | |
| . . | . . . . | | .\
| . . | . . . . | | . 1. i : Id
| . . | . . . . | | . 2. k : Knd
| . . | . . . . | | . 3. l : IdLnk
| . . | . . . . | | . 4. ds : x:Id fp-> Type
| . . | . . . . | | . 5. da : a:Knd fp-> Type
| . . | . . . . | | . 6. f :
| . . | . . . . | | . |  (tg:Id
| . . | . . . . | | . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | | . |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . | | . 7. source(l) = i
| . . | . . . . | | . 8. D' : Dsys
| . . | . . . . | | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
```

```
| . . | . . . . . | | . 10. w : World
| . . | . . . . . | | . 11. p : FairFifo
| . . | . . . . . | | . 12. PossibleWorld(D';w)
| . . | . . . . . | | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | | . 14. ∀e:E.
| . . | . . . . . | | . |   loc(e) = i
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . 15. ∀e:E.
| . . | . . . . . | | . |   isrcv(e)
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   lnk(e) = l
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | | . 17. ∀e:E.
| . . | . . . . . | | . |   loc(e) = i
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   kind(e) = k
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   sends(l;e)
| . . | . . . . . | | . |   =
| . . | . . . . . | | . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | | . 19. ∀e:E.
| . . | . . . . . | | . |   loc(e) = i
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . 20. ∀e:E.
| . . | . . . . . | | . |   isrcv(e)
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   lnk(e) = l
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . 21. e : E
| . . | . . . . . | | . 22. loc(e) = i
| . . | . . . . . | | . 23. kind(e) = k
| . . | . . . . . | | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . 25. sends(l;e)
| . . | . . . . . | | . |   =
| . . | . . . . . | | . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . 27. L :
| . . | . . . . . | | . |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | | . |   ∃e':E.
| . . | . . . . . | | . |   isrcv(e')
| . . | . . . . . | | . |   & lnk(e') = l
| . . | . . . . . | | . |     & sender(e') = e
| . . | . . . . . | | . |     & index(e') = n
| . . | . . . . . | | . 28. LL :
| . . | . . . . . | | . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . . | | . |                        | isrcv(e')
| . . | . . . . . | | . |                          & lnk(e') = l
| . . | . . . . . | | . |                            & sender(e') = e
```

```
| . . | . . . . | | . |                            & index(e') = n }
| . . | . . . . | | . 29. (λn.1of(L(n))) = LL
| . . | . . . . | | . 30. ∀e:E.
| . . | . . . . | | . |   isrcv(e)
| . . | . . . . | | . |   ⇒
| . . | . . . . | | . |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | | . |   =
| . . | . . . . | | . |   msg(lnk(e);tag(e);val(e))
| . . | . . . . | | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | | . |   ∃e':E.
| . . | . . . . | | . |   isrcv(e')
| . . | . . . . | | . |   & lnk(e') = l
| . . | . . . . | | . |     & sender(e') = e
| . . | . . . . | | . |     & index(e') = n
| . . | . . . . | | . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | | . |                         ×ma-valtype(da;
| . . | . . . . | | . |                                     rcv(l; tg)
| . . | . . . . | | . |                                     )))
| . . | . . . . | | . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | | . |   ∈ {e':E
| . . | . . . . | | . |      | isrcv(e')
| . . | . . . . | | . |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . | | . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | | . 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | | . |   =
| . . | . . . . | | . |   ||sends(l;e)||
| . . | . . . . | | . 36. ||map(λe'.
| . . | . . . . | | . |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | . |   =
| . . | . . . . | | . |   ||sends(l;e)||
| . . | . . . . | | . 37. i1 : ℕ
| . . | . . . . | | . 38. i1<||map(λe'.
| . . | . . . . | | . |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | . 39. v : E
| . . | . . . . | | . 40. isrcv(v)
| . . | . . . . | | . 41. lnk(v) = l
| . . | . . . . | | . 42. sender(v) = e
| . . | . . . . | | . 43. index(v) = i1
| . . | . . . . | | . 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | | . |   ∃e':E.
| . . | . . . . | | . |   isrcv(e')
| . . | . . . . | | . |   & lnk(e') = l
| . . | . . . . | | . |     & sender(e') = v
| . . | . . . . | | . |     & index(e') = n
| . . | . . . . | | . 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | . |   =
| . . | . . . . | | . |   msg(lnk(v);tag(v);val(v))
| . . | . . . . | | . 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | . 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | . 48. z : Msg
| . . | . . . . | | . 49. z = sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | . ⊢  (z = sends(l;e)[i1]) ∈ Prop  by Auto
| . . | . . . . | | . |
| . . | . . . . | | . .....subterm .....
| . . | . . . . | | . 1. i : Id
```

```
| . . | . . . . . | | . 2. k : Knd
| . . | . . . . . | | . 3. l : IdLnk
| . . | . . . . . | | . 4. ds : x:Id fp-> Type
| . . | . . . . . | | . 5. da : a:Knd fp-> Type
| . . | . . . . . | | . 6. f :
| . . | . . . . . | | . |  (tg:Id
| . . | . . . . . | | . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . . | | . |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . . | | . 7. source(l) = i
| . . | . . . . . | | . 8. D' : Dsys
| . . | . . . . . | | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . . | | . 10. w : World
| . . | . . . . . | | . 11. p : FairFifo
| . . | . . . . . | | . 12. PossibleWorld(D';w)
| . . | . . . . . | | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | | . 14. ∀e:E.
| . . | . . . . . | | . |   loc(e) = i
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . 15. ∀e:E.
| . . | . . . . . | | . |   isrcv(e)
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   lnk(e) = l
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | | . 17. ∀e:E.
| . . | . . . . . | | . |   loc(e) = i
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   kind(e) = k
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   sends(l;e)
| . . | . . . . . | | . |   =
| . . | . . . . . | | . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | | . 19. ∀e:E.
| . . | . . . . . | | . |   loc(e) = i
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . 20. ∀e:E.
| . . | . . . . . | | . |   isrcv(e)
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   lnk(e) = l
| . . | . . . . . | | . |   ⇒
| . . | . . . . . | | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . 21. e : E
| . . | . . . . . | | . 22. loc(e) = i
| . . | . . . . . | | . 23. kind(e) = k
| . . | . . . . . | | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . 25. sends(l;e)
| . . | . . . . . | | . |   =
| . . | . . . . . | | . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . 27. L :
```

```
| . . | . . . . | | . |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . | | . |   ∃e':E.
| . . | . . . . | | . |   isrcv(e')
| . . | . . . . | | . |   & lnk(e') = l
| . . | . . . . | | . |     & sender(e') = e
| . . | . . . . | | . |     & index(e') = n
| . . | . . . . | | . 28. LL :
| . . | . . . . | | . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | | . |                     | isrcv(e')
| . . | . . . . | | . |                       & lnk(e') = l
| . . | . . . . | | . |                         & sender(e') = e
| . . | . . . . | | . |                         & index(e') = n }
| . . | . . . . | | . 29. (λn.1of(L(n))) = LL
| . . | . . . . | | . 30. ∀e:E.
| . . | . . . . | | . |   isrcv(e)
| . . | . . . . | | . |   ⇒
| . . | . . . . | | . |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | | . |   =
| . . | . . . . | | . |   msg(lnk(e);tag(e);val(e))
| . . | . . . . | | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | | . |   ∃e':E.
| . . | . . . . | | . |   isrcv(e')
| . . | . . . . | | . |   & lnk(e') = l
| . . | . . . . | | . |     & sender(e') = e
| . . | . . . . | | . |     & index(e') = n
| . . | . . . . | | . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | | . |                       ×ma-valtype(da;
| . . | . . . . | | . |                                   rcv(l; tg)
| . . | . . . . | | . |                                   )))
| . . | . . . . | | . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | | . |   ∈ {e':E
| . . | . . . . | | . |      | isrcv(e')
| . . | . . . . | | . |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . | | . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | | . 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | | . |   =
| . . | . . . . | | . |   ||sends(l;e)||
| . . | . . . . | | . 36. ||map(λe'.
| . . | . . . . | | . |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | . |   =
| . . | . . . . | | . |   ||sends(l;e)||
| . . | . . . . | | . 37. i1 : ℕ
| . . | . . . . | | . 38. i1<||map(λe'.
| . . | . . . . | | . |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | . 39. v : E
| . . | . . . . | | . 40. isrcv(v)
| . . | . . . . | | . 41. lnk(v) = l
| . . | . . . . | | . 42. sender(v) = e
| . . | . . . . | | . 43. index(v) = i1
| . . | . . . . | | . 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | | . |   ∃e':E.
| . . | . . . . | | . |   isrcv(e')
| . . | . . . . | | . |   & lnk(e') = l
| . . | . . . . | | . |     & sender(e') = v
| . . | . . . . | | . |     & index(e') = n
```

```
| . . | . . . . | | . 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | . |   =
| . . | . . . . | | . |   msg(lnk(v);tag(v);val(v))
| . . | . . . . | | . 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | . 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | . 48. z : Msg
| . . | . . . . | | . 49. z = sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | . ⊢  z ∈ IdLnk×(tg:Id×ma-valtype(da; rcv(l; tg)))
| . . | . . . . | | . | by Assert (z ∈ (Msg on l))
| . . | . . . . | | . |
| . . | . . . . | | . |\
| . . | . . . . | | . | .....assertion .....
| . . | . . . . | | . | 1. i : Id
| . . | . . . . | | . | 2. k : Knd
| . . | . . . . | | . | 3. l : IdLnk
| . . | . . . . | | . | 4. ds : x:Id fp-> Type
| . . | . . . . | | . | 5. da : a:Knd fp-> Type
| . . | . . . . | | . | 6. f :
| . . | . . . . | | . | |  (tg:Id
| . . | . . . . | | . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | | . | |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . | | . | 7. source(l) = i
| . . | . . . . | | . | 8. D' : Dsys
| . . | . . . . | | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | | . | 10. w : World
| . . | . . . . | | . | 11. p : FairFifo
| . . | . . . . | | . | 12. PossibleWorld(D';w)
| . . | . . . . | | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | . | 14. ∀e:E.
| . . | . . . . | | . | |   loc(e) = i
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | . | 15. ∀e:E.
| . . | . . . . | | . | |   isrcv(e)
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   lnk(e) = l
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . | | . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . | | . | 17. ∀e:E.
| . . | . . . . | | . | |   loc(e) = i
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   kind(e) = k
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   sends(l;e)
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | . | 19. ∀e:E.
| . . | . . . . | | . | |   loc(e) = i
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | . | 20. ∀e:E.
| . . | . . . . | | . | |   isrcv(e)
```

```
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   lnk(e) = l
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | . | 21. e : E
| . . | . . . . | | . | 22. loc(e) = i
| . . | . . . . | | . | 23. kind(e) = k
| . . | . . . . | | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | . | 25. sends(l;e)
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | . | 27. L :
| . . | . . . . | | . | |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . | | . | |   ∃e':E.
| . . | . . . . | | . | |   isrcv(e')
| . . | . . . . | | . | |   & lnk(e') = l
| . . | . . . . | | . | |     & sender(e') = e
| . . | . . . . | | . | |     & index(e') = n
| . . | . . . . | | . | 28. LL :
| . . | . . . . | | . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | | . | |                        | isrcv(e')
| . . | . . . . | | . | |                          & lnk(e') = l
| . . | . . . . | | . | |                            & sender(e') = e
| . . | . . . . | | . | |                            & index(e') = n }
| . . | . . . . | | . | 29. (λn.1of(L(n))) = LL
| . . | . . . . | | . | 30. ∀e:E.
| . . | . . . . | | . | |   isrcv(e)
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   msg(lnk(e);tag(e);val(e))
| . . | . . . . | | . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | | . | |   ∃e':E.
| . . | . . . . | | . | |   isrcv(e')
| . . | . . . . | | . | |   & lnk(e') = l
| . . | . . . . | | . | |     & sender(e') = e
| . . | . . . . | | . | |     & index(e') = n
| . . | . . . . | | . | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | | . | |                          ×ma-valtype(da;
| . . | . . . . | | . | |                                      rcv(l;
| . . | . . . . | | . | |                                         tg)
| . . | . . . . | | . | |                                      )))
| . . | . . . . | | . | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | | . | |   ∈ {e':E
| . . | . . . . | | . | |      | isrcv(e')
| . . | . . . . | | . | |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . | | . | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | | . | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   ||sends(l;e)||
| . . | . . . . | | . | 36. ||map(λe'.
| . . | . . . . | | . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   ||sends(l;e)||
```

```
| . . | . . . . . | | . | 37. i1 : ℕ
| . . | . . . . . | | . | 38. i1<||map(λe'.
| . . | . . . . . | | . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | | . | 39. v : E
| . . | . . . . . | | . | 40. isrcv(v)
| . . | . . . . . | | . | 41. lnk(v) = l
| . . | . . . . . | | . | 42. sender(v) = e
| . . | . . . . . | | . | 43. index(v) = i1
| . . | . . . . . | | . | 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . . | | . | |   ∃e':E.
| . . | . . . . . | | . | |   isrcv(e')
| . . | . . . . . | | . | |   & lnk(e') = l
| . . | . . . . . | | . | |     & sender(e') = v
| . . | . . . . . | | . | |     & index(e') = n
| . . | . . . . . | | . | 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . . | | . | |   =
| . . | . . . . . | | . | |   msg(lnk(v);tag(v);val(v))
| . . | . . . . . | | . | 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | | . | 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | | . | 48. z : Msg
| . . | . . . . . | | . | 49. z = sends(lnk(v);sender(v))[index(v)]
| . . | . . . . . | | . | ⊢  z ∈ (Msg on l)
| . . | . . . . . | | . | | by At Type Analyze THEN Unfold `haslink` 0
| . . | . . . . . | | . | |
| . . | . . . . . | | . | 1. i : Id
| . . | . . . . . | | . | 2. k : Knd
| . . | . . . . . | | . | 3. l : IdLnk
| . . | . . . . . | | . | 4. ds : x:Id fp-> Type
| . . | . . . . . | | . | 5. da : a:Knd fp-> Type
| . . | . . . . . | | . | 6. f :
| . . | . . . . . | | . | |  (tg:Id
| . . | . . . . . | | . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . . | | . | |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . . | | . | 7. source(l) = i
| . . | . . . . . | | . | 8. D' : Dsys
| . . | . . . . . | | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . . | | . | 10. w : World
| . . | . . . . . | | . | 11. p : FairFifo
| . . | . . . . . | | . | 12. PossibleWorld(D';w)
| . . | . . . . . | | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | | . | 14. ∀e:E.
| . . | . . . . . | | . | |   loc(e) = i
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . | 15. ∀e:E.
| . . | . . . . . | | . | |   isrcv(e)
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   lnk(e) = l
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | | . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | | . | 17. ∀e:E.
| . . | . . . . . | | . | |   loc(e) = i
| . . | . . . . . | | . | |   ⇒
```

```
| . . | . . . . | | . | |   kind(e) = k
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   sends(l;e)
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | . | 19. ∀e:E.
| . . | . . . . | | . | |   loc(e) = i
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | . | 20. ∀e:E.
| . . | . . . . | | . | |   isrcv(e)
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   lnk(e) = l
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | . | 21. e : E
| . . | . . . . | | . | 22. loc(e) = i
| . . | . . . . | | . | 23. kind(e) = k
| . . | . . . . | | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | . | 25. sends(l;e)
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . | | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . | | . | 27. L :
| . . | . . . . | | . | |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . | | . | |   ∃e':E.
| . . | . . . . | | . | |   isrcv(e')
| . . | . . . . | | . | |   & lnk(e') = l
| . . | . . . . | | . | |     & sender(e') = e
| . . | . . . . | | . | |     & index(e') = n
| . . | . . . . | | . | 28. LL :
| . . | . . . . | | . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | | . | |                      | isrcv(e')
| . . | . . . . | | . | |                        & lnk(e') = l
| . . | . . . . | | . | |                          & sender(e') = e
| . . | . . . . | | . | |                          & index(e') = n }

| . . | . . . . | | . | 29. (λn.1of(L(n))) = LL
| . . | . . . . | | . | 30. ∀e:E.
| . . | . . . . | | . | |   isrcv(e)
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   msg(lnk(e);tag(e);val(e))
| . . | . . . . | | . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | | . | |   ∃e':E.
| . . | . . . . | | . | |   isrcv(e')
| . . | . . . . | | . | |   & lnk(e') = l
| . . | . . . . | | . | |     & sender(e') = e
| . . | . . . . | | . | |     & index(e') = n
| . . | . . . . | | . | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | | . | |                         ×ma-valtype(da;
| . . | . . . . | | . | |                                     rcv(l;
| . . | . . . . | | . | |                                         tg)
| . . | . . . . | | . | |                                     )))
```

```
| . . | . . . . | | . | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | | . | |   ∈ {e':E
| . . | . . . . | | . | |      | isrcv(e')
| . . | . . . . | | . | |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . | | . | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | | . | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   ||sends(l;e)||
| . . | . . . . | | . | 36. ||map(λe'.
| . . | . . . . | | . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   ||sends(l;e)||
| . . | . . . . | | . | 37. i1 : ℕ
| . . | . . . . | | . | 38. i1<||map(λe'.
| . . | . . . . | | . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | . | 39. v : E
| . . | . . . . | | . | 40. isrcv(v)
| . . | . . . . | | . | 41. lnk(v) = l
| . . | . . . . | | . | 42. sender(v) = e
| . . | . . . . | | . | 43. index(v) = i1
| . . | . . . . | | . | 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | | . | |   ∃e':E.
| . . | . . . . | | . | |   isrcv(e')
| . . | . . . . | | . | |   & lnk(e') = l
| . . | . . . . | | . | |     & sender(e') = v
| . . | . . . . | | . | |     & index(e') = n
| . . | . . . . | | . | 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   msg(lnk(v);tag(v);val(v))
| . . | . . . . | | . | 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | . | 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | . | 48. z : Msg
| . . | . . . . | | . | 49. z = sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | . | ⊢  mlnk(z) = l  by HypSubst -1 0
| . . | . . . . | | . | |
| . . | . . . . | | . | 1. i : Id
| . . | . . . . | | . | 2. k : Knd
| . . | . . . . | | . | 3. l : IdLnk
| . . | . . . . | | . | 4. ds : x:Id fp-> Type
| . . | . . . . | | . | 5. da : a:Knd fp-> Type
| . . | . . . . | | . | 6. f :
| . . | . . . . | | . | |  (tg:Id
| . . | . . . . | | . | |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | | . | |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . | | . | 7. source(l) = i
| . . | . . . . | | . | 8. D' : Dsys
| . . | . . . . | | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | | . | 10. w : World
| . . | . . . . | | . | 11. p : FairFifo
| . . | . . . . | | . | 12. PossibleWorld(D';w)
| . . | . . . . | | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | . | 14. ∀e:E.
| . . | . . . . | | . | |   loc(e) = i
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
```

```
| . . | . . . . . | | . | 15. ∀e:E.
| . . | . . . . . | | . | |   isrcv(e)
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   lnk(e) = l
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | | . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | | . | 17. ∀e:E.
| . . | . . . . . | | . | |   loc(e) = i
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   kind(e) = k
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   sends(l;e)
| . . | . . . . . | | . | |   =
| . . | . . . . . | | . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | | . | 19. ∀e:E.
| . . | . . . . . | | . | |   loc(e) = i
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . | 20. ∀e:E.
| . . | . . . . . | | . | |   isrcv(e)
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   lnk(e) = l
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))

| . . | . . . . . | | . | 21. e : E
| . . | . . . . . | | . | 22. loc(e) = i
| . . | . . . . . | | . | 23. kind(e) = k
| . . | . . . . . | | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . | 25. sends(l;e)
| . . | . . . . . | | . | |   =
| . . | . . . . . | | . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . | 27. L :
| . . | . . . . . | | . | |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | | . | |   ∃e':E.
| . . | . . . . . | | . | |   isrcv(e')
| . . | . . . . . | | . | |   & lnk(e') = l
| . . | . . . . . | | . | |     & sender(e') = e
| . . | . . . . . | | . | |     & index(e') = n
| . . | . . . . . | | . | 28. LL :
| . . | . . . . . | | . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . . | | . | |                      | isrcv(e')
| . . | . . . . . | | . | |                        & lnk(e') = l
| . . | . . . . . | | . | |                          & sender(e') = e
| . . | . . . . . | | . | |                          & index(e') = n }
| . . | . . . . . | | . | 29. (λn.1of(L(n))) = LL
| . . | . . . . . | | . | 30. ∀e:E.
| . . | . . . . . | | . | |   isrcv(e)
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . | | . | |   =
| . . | . . . . . | | . | |   msg(lnk(e);tag(e);val(e))
```

```
| . . | . . . . . | | . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . | | . | |   ∃e':E.
| . . | . . . . . | | . | |   isrcv(e')
| . . | . . . . . | | . | |   & lnk(e') = l
| . . | . . . . . | | . | |     & sender(e') = e
| . . | . . . . . | | . | |     & index(e') = n
| . . | . . . . . | | . | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . . | | . | |                         ×ma-valtype(da;
| . . | . . . . . | | . | |                                     rcv(l;
| . . | . . . . . | | . | |                                         tg)
| . . | . . . . . | | . | |                                     )))
| . . | . . . . . | | . | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . . | | . | |   ∈ {e':E
| . . | . . . . . | | . | |      | isrcv(e')
| . . | . . . . . | | . | |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . . | | . | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . . | | . | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . . | | . | |   =
| . . | . . . . . | | . | |   ||sends(l;e)||
| . . | . . . . . | | . | 36. ||map(λe'.
| . . | . . . . . | | . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | | . | |   =
| . . | . . . . . | | . | |   ||sends(l;e)||
| . . | . . . . . | | . | 37. i1 : ℕ
| . . | . . . . . | | . | 38. i1<||map(λe'.
| . . | . . . . . | | . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | | . | 39. v : E
| . . | . . . . . | | . | 40. isrcv(v)
| . . | . . . . . | | . | 41. lnk(v) = l
| . . | . . . . . | | . | 42. sender(v) = e
| . . | . . . . . | | . | 43. index(v) = i1
| . . | . . . . . | | . | 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . . | | . | |   ∃e':E.
| . . | . . . . . | | . | |   isrcv(e')
| . . | . . . . . | | . | |   & lnk(e') = l
| . . | . . . . . | | . | |     & sender(e') = v
| . . | . . . . . | | . | |     & index(e') = n
| . . | . . . . . | | . | 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . . | | . | |   =
| . . | . . . . . | | . | |   msg(lnk(v);tag(v);val(v))
| . . | . . . . . | | . | 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | | . | 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | | . | 48. z : Msg
| . . | . . . . . | | . | 49. z = sends(lnk(v);sender(v))[index(v)]
| . . | . . . . . | | . | ⊢  mlnk(sends(lnk(v);sender(v))[index(v)]) = l
| . . | . . . . . | | . | | by HypSubst' -9 0
| . . | . . . . . | | . | |
| . . | . . . . . | | . | 1. i : Id
| . . | . . . . . | | . | 2. k : Knd
| . . | . . . . . | | . | 3. l : IdLnk
| . . | . . . . . | | . | 4. ds : x:Id fp-> Type
| . . | . . . . . | | . | 5. da : a:Knd fp-> Type
| . . | . . . . . | | . | 6. f :
| . . | . . . . . | | . | |  (tg:Id
| . . | . . . . . | | . | |   ×State(ds)→ma-valtype(da; k)→
```

```
| . . | . . . . . | | . | |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . . | | . | 7. source(l) = i
| . . | . . . . . | | . | 8. D' : Dsys
| . . | . . . . . | | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . . | | . | 10. w : World
| . . | . . . . . | | . | 11. p : FairFifo
| . . | . . . . . | | . | 12. PossibleWorld(D';w)
| . . | . . . . . | | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | | . | 14. ∀e:E.
| . . | . . . . . | | . | |   loc(e) = i
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . | 15. ∀e:E.
| . . | . . . . . | | . | |   isrcv(e)
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   lnk(e) = l
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | | . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | | . | 17. ∀e:E.
| . . | . . . . . | | . | |   loc(e) = i
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   kind(e) = k
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   sends(l;e)
| . . | . . . . . | | . | |   =
| . . | . . . . . | | . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | | . | 19. ∀e:E.
| . . | . . . . . | | . | |   loc(e) = i
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . | 20. ∀e:E.
| . . | . . . . . | | . | |   isrcv(e)
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   lnk(e) = l
| . . | . . . . . | | . | |   ⇒
| . . | . . . . . | | . | |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . | 21. e : E
| . . | . . . . . | | . | 22. loc(e) = i
| . . | . . . . . | | . | 23. kind(e) = k
| . . | . . . . . | | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . | 25. sends(l;e)
| . . | . . . . . | | . | |   =
| . . | . . . . . | | . | |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . | 27. L :
| . . | . . . . . | | . | |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | | . | |   ∃e':E.
| . . | . . . . . | | . | |   isrcv(e')
| . . | . . . . . | | . | |   & lnk(e') = l
| . . | . . . . . | | . | |     & sender(e') = e
| . . | . . . . . | | . | |     & index(e') = n
| . . | . . . . . | | . | 28. LL :
```

```
| . . | . . . . | | . | |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . | | . | |                       | isrcv(e')
| . . | . . . . | | . | |                         & lnk(e') = l
| . . | . . . . | | . | |                           & sender(e') = e
| . . | . . . . | | . | |                           & index(e') = n }
| . . | . . . . | | . | 29. (λn.1of(L(n))) = LL
| . . | . . . . | | . | 30. ∀e:E.
| . . | . . . . | | . | |   isrcv(e)
| . . | . . . . | | . | |   ⇒
| . . | . . . . | | . | |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   msg(lnk(e);tag(e);val(e))
| . . | . . . . | | . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . | | . | |   ∃e':E.
| . . | . . . . | | . | |   isrcv(e')
| . . | . . . . | | . | |   & lnk(e') = l
| . . | . . . . | | . | |     & sender(e') = e
| . . | . . . . | | . | |     & index(e') = n
| . . | . . . . | | . | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . | | . | |                         ×ma-valtype(da;
| . . | . . . . | | . | |                                     rcv(l;
| . . | . . . . | | . | |                                         tg)
| . . | . . . . | | . | |                                     )))
| . . | . . . . | | . | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | | . | |   ∈ {e':E
| . . | . . . . | | . | |      | isrcv(e')
| . . | . . . . | | . | |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . | | . | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | | . | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   ||sends(l;e)||
| . . | . . . . | | . | 36. ||map(λe'.
| . . | . . . . | | . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   ||sends(l;e)||
| . . | . . . . | | . | 37. i1 : ℕ
| . . | . . . . | | . | 38. i1<||map(λe'.
| . . | . . . . | | . | |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | | . | 39. v : E
| . . | . . . . | | . | 40. isrcv(v)
| . . | . . . . | | . | 41. lnk(v) = l
| . . | . . . . | | . | 42. sender(v) = e
| . . | . . . . | | . | 43. index(v) = i1
| . . | . . . . | | . | 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | | . | |   ∃e':E.
| . . | . . . . | | . | |   isrcv(e')
| . . | . . . . | | . | |   & lnk(e') = l
| . . | . . . . | | . | |     & sender(e') = v
| . . | . . . . | | . | |     & index(e') = n
| . . | . . . . | | . | 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | . | |   =
| . . | . . . . | | . | |   msg(lnk(v);tag(v);val(v))
| . . | . . . . | | . | 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | . | 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | . | 48. z : Msg
```

```
| . . | . . . . | | . | 49. z = sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | . | ⊢  mlnk(sends(l;sender(v))[index(v)]) = l
| . . | . . . . | | . | | by GenConclAtAddr [2;1]
| . . | . . . . | | . | |
| . . | . . . . | | . | |\
| . . | . . . . | | . | | 1. i : Id
| . . | . . . . | | . | | 2. k : Knd
| . . | . . . . | | . | | 3. l : IdLnk
| . . | . . . . | | . | | 4. ds : x:Id fp-> Type
| . . | . . . . | | . | | 5. da : a:Knd fp-> Type
| . . | . . . . | | . | | 6. f :
| . . | . . . . | | . | |    (tg:Id
| . . | . . . . | | . | |     ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | | . | |     (da(rcv(l; tg))?Void List)) List
| . . | . . . . | | . | | 7. source(l) = i
| . . | . . . . | | . | | 8. D' : Dsys
| . . | . . . . | | . | | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | | . | | 10. w : World
| . . | . . . . | | . | | 11. p : FairFifo
| . . | . . . . | | . | | 12. PossibleWorld(D';w)
| . . | . . . . | | . | | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | . | | 14. ∀e:E.
| . . | . . . . | | . | |     loc(e) = i
| . . | . . . . | | . | |     ⇒
| . . | . . . . | | . | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | . | | 15. ∀e:E.
| . . | . . . . | | . | |     isrcv(e)
| . . | . . . . | | . | |     ⇒
| . . | . . . . | | . | |     lnk(e) = l
| . . | . . . . | | . | |     ⇒
| . . | . . . . | | . | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | . | | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . | | . | |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . | | . | | 17. ∀e:E.
| . . | . . . . | | . | |     loc(e) = i
| . . | . . . . | | . | |     ⇒
| . . | . . . . | | . | |     kind(e) = k
| . . | . . . . | | . | |     ⇒
| . . | . . . . | | . | |     sends(l;e)
| . . | . . . . | | . | |     =
| . . | . . . . | | . | |     tagged-messages(l;λz.
| . . | . . . . | | . | |     (z when e);val(e);f)
| . . | . . . . | | . | | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | | . | | 19. ∀e:E.
| . . | . . . . | | . | |     loc(e) = i
| . . | . . . . | | . | |     ⇒
| . . | . . . . | | . | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | . | | 20. ∀e:E.
| . . | . . . . | | . | |     isrcv(e)
| . . | . . . . | | . | |     ⇒
| . . | . . . . | | . | |     lnk(e) = l
| . . | . . . . | | . | |     ⇒
| . . | . . . . | | . | |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | | . | | 21. e : E
| . . | . . . . | | . | | 22. loc(e) = i
```

```
| . . | . . . . . | | . | | 23. kind(e) = k
| . . | . . . . . | | . | | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . | | 25. sends(l;e)
| . . | . . . . . | | . | |     =
| . . | . . . . . | | . | |     tagged-messages(l;λz.
| . . | . . . . . | | . | |     (z when e);val(e);f)
| . . | . . . . . | | . | | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . | | 27. L :
| . . | . . . . . | | . | |     ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | | . | |     ∃e':E.
| . . | . . . . . | | . | |     isrcv(e')
| . . | . . . . . | | . | |     & lnk(e') = l
| . . | . . . . . | | . | |       & sender(e') = e
| . . | . . . . . | | . | |       & index(e') = n
| . . | . . . . . | | . | | 28. LL :
| . . | . . . . . | | . | |     n:ℕ||sends(l;e)||→
| . . | . . . . . | | . | |     {e':E
| . . | . . . . . | | . | |      | isrcv(e')
| . . | . . . . . | | . | |        & lnk(e') = l
| . . | . . . . . | | . | |          & sender(e') = e
| . . | . . . . . | | . | |          & index(e') = n }
| . . | . . . . . | | . | | 29. (λn.1of(L(n))) = LL
| . . | . . . . . | | . | | 30. ∀e:E.
| . . | . . . . . | | . | |     isrcv(e)
| . . | . . . . . | | . | |     ⇒
| . . | . . . . . | | . | |     sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . | | . | |     =
| . . | . . . . . | | . | |     msg(lnk(e);tag(e);val(e))
| . . | . . . . . | | . | | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . | | . | |     ∃e':E.
| . . | . . . . . | | . | |     isrcv(e')
| . . | . . . . . | | . | |     & lnk(e') = l
| . . | . . . . . | | . | |       & sender(e') = e
| . . | . . . . . | | . | |       & index(e') = n
| . . | . . . . . | | . | | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . . | | . | |                           ×ma-valtype(da;
| . . | . . . . . | | . | |                                       rcv(l;
| . . | . . . . . | | . | |                                           tg
| . . | . . . . . | | . | |                                           )
| . . | . . . . . | | . | |                                       )))
| . . | . . . . . | | . | | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . . | | . | |     ∈ {e':E
| . . | . . . . . | | . | |         | isrcv(e')
| . . | . . . . . | | . | |           & lnk(e') = l
| . . | . . . . . | | . | |             & sender(e') = e } List
| . . | . . . . . | | . | | 34. ||upto(||sends(l;e)||)||
| . . | . . . . . | | . | |     =
| . . | . . . . . | | . | |     ||sends(l;e)||
| . . | . . . . . | | . | | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . . | | . | |     =
| . . | . . . . . | | . | |     ||sends(l;e)||
| . . | . . . . . | | . | | 36. ||map(λe'.

| . . | . . . . . | | . | |     <tag(e'),val(e')>;map(LL;upto(...)))||
| . . | . . . . . | | . | |     =
| . . | . . . . . | | . | |     ||sends(l;e)||
```

```
| . . | . . . . . | | . | | 37. i1 : ℕ
| . . | . . . . . | | . | | 38. i1<||map(λe'.
| . . | . . . . . | | . | |     <tag(e'),val(e')>;map(LL;upto(...)))||
| . . | . . . . . | | . | | 39. v : E
| . . | . . . . . | | . | | 40. isrcv(v)
| . . | . . . . . | | . | | 41. lnk(v) = l
| . . | . . . . . | | . | | 42. sender(v) = e
| . . | . . . . . | | . | | 43. index(v) = i1
| . . | . . . . . | | . | | 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . . | | . | |     ∃e':E.
| . . | . . . . . | | . | |     isrcv(e')
| . . | . . . . . | | . | |     & lnk(e') = l
| . . | . . . . . | | . | |       & sender(e') = v
| . . | . . . . . | | . | |       & index(e') = n
| . . | . . . . . | | . | | 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . . | | . | |     =
| . . | . . . . . | | . | |     msg(lnk(v);tag(v);val(v))
| . . | . . . . . | | . | | 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | | . | | 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | | . | | 48. z : Msg
| . . | . . . . . | | . | | 49. z = sends(lnk(v);sender(v))[index(v)]
| . . | . . . . . | | . | | ⊢  index(v)<||sends(l;sender(v))||
| . . | . . . . . | | . | |   by HypSubst -8 0
| . . | . . . . . | | . | |
| . . | . . . . . | | . | .\
| . . | . . . . . | | . | . 1. i : Id
| . . | . . . . . | | . | . 2. k : Knd
| . . | . . . . . | | . | . 3. l : IdLnk
| . . | . . . . . | | . | . 4. ds : x:Id fp-> Type
| . . | . . . . . | | . | . 5. da : a:Knd fp-> Type
| . . | . . . . . | | . | . 6. f :
| . . | . . . . . | | . | .    (tg:Id
| . . | . . . . . | | . | .     ×State(ds)→ma-valtype(da; k)→
| . . | . . . . . | | . | .     (da(rcv(l; tg))?Void List)) List
| . . | . . . . . | | . | . 7. source(l) = i
| . . | . . . . . | | . | . 8. D' : Dsys
| . . | . . . . . | | . | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . . | | . | . 10. w : World
| . . | . . . . . | | . | . 11. p : FairFifo
| . . | . . . . . | | . | . 12. PossibleWorld(D';w)
| . . | . . . . . | | . | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | | . | . 14. ∀e:E.
| . . | . . . . . | | . | .     loc(e) = i
| . . | . . . . . | | . | .     ⇒
| . . | . . . . . | | . | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . | . 15. ∀e:E.
| . . | . . . . . | | . | .     isrcv(e)
| . . | . . . . . | | . | .     ⇒
| . . | . . . . . | | . | .     lnk(e) = l
| . . | . . . . . | | . | .     ⇒
| . . | . . . . . | | . | .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | | . | .     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | | . | . 17. ∀e:E.
| . . | . . . . . | | . | .     loc(e) = i
```

```
| . . | . . . . . | | . | .      ⇒
| . . | . . . . . | | . | .      kind(e) = k
| . . | . . . . . | | . | .      ⇒
| . . | . . . . . | | . | .      sends(l;e)
| . . | . . . . . | | . | .      =
| . . | . . . . . | | . | .      tagged-messages(l;λz.
| . . | . . . . . | | . | .      (z when e);val(e);f)
| . . | . . . . . | | . | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | | . | . 19. ∀e:E.
| . . | . . . . . | | . | .      loc(e) = i
| . . | . . . . . | | . | .      ⇒
| . . | . . . . . | | . | .      (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . | . 20. ∀e:E.
| . . | . . . . . | | . | .      isrcv(e)
| . . | . . . . . | | . | .      ⇒
| . . | . . . . . | | . | .      lnk(e) = l
| . . | . . . . . | | . | .      ⇒
| . . | . . . . . | | . | .      (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . | . 21. e : E
| . . | . . . . . | | . | . 22. loc(e) = i
| . . | . . . . . | | . | . 23. kind(e) = k
| . . | . . . . . | | . | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . | . 25. sends(l;e)
| . . | . . . . . | | . | .      =
| . . | . . . . . | | . | .      tagged-messages(l;λz.
| . . | . . . . . | | . | .      (z when e);val(e);f)
| . . | . . . . . | | . | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . | . 27. L :
| . . | . . . . . | | . | .      ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | | . | .      ∃e':E.
| . . | . . . . . | | . | .      isrcv(e')
| . . | . . . . . | | . | .      & lnk(e') = l
| . . | . . . . . | | . | .        & sender(e') = e
| . . | . . . . . | | . | .        & index(e') = n
| . . | . . . . . | | . | . 28. LL :
| . . | . . . . . | | . | .      n:ℕ||sends(l;e)||→
| . . | . . . . . | | . | .      {e':E
| . . | . . . . . | | . | .       | isrcv(e')
| . . | . . . . . | | . | .         & lnk(e') = l
| . . | . . . . . | | . | .           & sender(e') = e
| . . | . . . . . | | . | .           & index(e') = n }
| . . | . . . . . | | . | . 29. (λn.1of(L(n))) = LL

| . . | . . . . . | | . | . 30. ∀e:E.
| . . | . . . . . | | . | .      isrcv(e)
| . . | . . . . . | | . | .      ⇒
| . . | . . . . . | | . | .      sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . | | . | .      =
| . . | . . . . . | | . | .      msg(lnk(e);tag(e);val(e))
| . . | . . . . . | | . | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . | | . | .      ∃e':E.
| . . | . . . . . | | . | .      isrcv(e')
| . . | . . . . . | | . | .      & lnk(e') = l
| . . | . . . . . | | . | .        & sender(e') = e
| . . | . . . . . | | . | .        & index(e') = n
| . . | . . . . . | | . | . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
```

```
| . . | . . . . | | . | .                           ×ma-valtype(da;
| . . | . . . . | | . | .                                       rcv(l;
| . . | . . . . | | . | .                                           tg
| . . | . . . . | | . | .                                           )
| . . | . . . . | | . | .                                       )))
| . . | . . . . | | . | . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . | | . | .     ∈ {e':E
| . . | . . . . | | . | .        | isrcv(e')
| . . | . . . . | | . | .          & lnk(e') = l
| . . | . . . . | | . | .            & sender(e') = e } List
| . . | . . . . | | . | . 34. ||upto(||sends(l;e)||)||
| . . | . . . . | | . | .     =
| . . | . . . . | | . | .     ||sends(l;e)||
| . . | . . . . | | . | . 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | | . | .     =
| . . | . . . . | | . | .     ||sends(l;e)||
| . . | . . . . | | . | . 36. ||map(λe'.
| . . | . . . . | | . | .     <tag(e'),val(e')>;map(LL;upto(...)))||
| . . | . . . . | | . | .     =
| . . | . . . . | | . | .     ||sends(l;e)||
| . . | . . . . | | . | . 37. i1 : ℕ
| . . | . . . . | | . | . 38. i1<||map(λe'.
| . . | . . . . | | . | .     <tag(e'),val(e')>;map(LL;upto(...)))||
| . . | . . . . | | . | . 39. v : E
| . . | . . . . | | . | . 40. isrcv(v)
| . . | . . . . | | . | . 41. lnk(v) = l
| . . | . . . . | | . | . 42. sender(v) = e
| . . | . . . . | | . | . 43. index(v) = i1
| . . | . . . . | | . | . 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | | . | .     ∃e':E.
| . . | . . . . | | . | .     isrcv(e')
| . . | . . . . | | . | .     & lnk(e') = l
| . . | . . . . | | . | .       & sender(e') = v
| . . | . . . . | | . | .       & index(e') = n
| . . | . . . . | | . | . 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | . | .     =
| . . | . . . . | | . | .     msg(lnk(v);tag(v);val(v))
| . . | . . . . | | . | . 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | . | . 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | | . | . 48. z : Msg
| . . | . . . . | | . | . 49. z = sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | | . | . 50. v1 : (Msg on l)
| . . | . . . . | | . | . 51. sends(l;sender(v))[index(v)] = v1
| . . | . . . . | | . | . ⊢  mlnk(v1) = l
| . . | . . . . | | . | .   by Analyze -2 THEN Unfold `haslink` -2
| . . | . . . . | | . |
| . . | . . . . | | . .\
| . . | . . . . | | . . 1. i : Id
| . . | . . . . | | . . 2. k : Knd
| . . | . . . . | | . . 3. l : IdLnk
| . . | . . . . | | . . 4. ds : x:Id fp-> Type
| . . | . . . . | | . . 5. da : a:Knd fp-> Type
| . . | . . . . | | . . 6. f :
| . . | . . . . | | . .    (tg:Id
| . . | . . . . | | . .     ×State(ds)→ma-valtype(da; k)→
```

```
| . . | . . . . . | | . .     (da(rcv(l; tg))?Void List)) List
| . . | . . . . . | | . . 7. source(l) = i
| . . | . . . . . | | . . 8. D' : Dsys
| . . | . . . . . | | . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . . | | . . 10. w : World
| . . | . . . . . | | . . 11. p : FairFifo
| . . | . . . . . | | . . 12. PossibleWorld(D';w)
| . . | . . . . . | | . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | | . . 14. ∀e:E.
| . . | . . . . . | | . .     loc(e) = i
| . . | . . . . . | | . .     ⇒
| . . | . . . . . | | . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . . 15. ∀e:E.
| . . | . . . . . | | . .     isrcv(e)
| . . | . . . . . | | . .     ⇒
| . . | . . . . . | | . .     lnk(e) = l
| . . | . . . . . | | . .     ⇒
| . . | . . . . . | | . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | | . .     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | | . . 17. ∀e:E.
| . . | . . . . . | | . .     loc(e) = i
| . . | . . . . . | | . .     ⇒
| . . | . . . . . | | . .     kind(e) = k
| . . | . . . . . | | . .     ⇒
| . . | . . . . . | | . .     sends(l;e)
| . . | . . . . . | | . .     =
| . . | . . . . . | | . .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | | . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | | . . 19. ∀e:E.
| . . | . . . . . | | . .     loc(e) = i
| . . | . . . . . | | . .     ⇒
| . . | . . . . . | | . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . . 20. ∀e:E.
| . . | . . . . . | | . .     isrcv(e)
| . . | . . . . . | | . .     ⇒
| . . | . . . . . | | . .     lnk(e) = l
| . . | . . . . . | | . .     ⇒
| . . | . . . . . | | . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | | . . 21. e : E
| . . | . . . . . | | . . 22. loc(e) = i
| . . | . . . . . | | . . 23. kind(e) = k
| . . | . . . . . | | . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . . 25. sends(l;e)
| . . | . . . . . | | . .     =
| . . | . . . . . | | . .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | | . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | | . . 27. L :
| . . | . . . . . | | . .     ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | | . .     ∃e':E.
| . . | . . . . . | | . .     isrcv(e')
| . . | . . . . . | | . .     & lnk(e') = l
| . . | . . . . . | | . .       & sender(e') = e
| . . | . . . . . | | . .       & index(e') = n
| . . | . . . . . | | . . 28. LL :
```

```
| . . | . . . . . | | . .     n:ℕ||sends(l;e)||→{e':E
| . . | . . . . . | | . .                          | isrcv(e')
| . . | . . . . . | | . .                            & lnk(e') = l
| . . | . . . . . | | . .                              & sender(e') = e
| . . | . . . . . | | . .                              & index(e') = n }
| . . | . . . . . | | . . 29. (λn.1of(L(n))) = LL
| . . | . . . . . | | . . 30. ∀e:E.
| . . | . . . . . | | . .     isrcv(e)
| . . | . . . . . | | . .     ⇒
| . . | . . . . . | | . .     sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . | | . .     =
| . . | . . . . . | | . .     msg(lnk(e);tag(e);val(e))
| . . | . . . . . | | . . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . | | . .     ∃e':E.
| . . | . . . . . | | . .     isrcv(e')
| . . | . . . . . | | . .     & lnk(e') = l
| . . | . . . . . | | . .       & sender(e') = e
| . . | . . . . . | | . .       & index(e') = n
| . . | . . . . . | | . . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . . | | . .                           ×ma-valtype(da;
| . . | . . . . . | | . .                                        rcv(l;
| . . | . . . . . | | . .                                            tg)
| . . | . . . . . | | . .                                        )))
| . . | . . . . . | | . . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . . | | . .     ∈ {e':E
| . . | . . . . . | | . .        | isrcv(e')
| . . | . . . . . | | . .          & lnk(e') = l & sender(e') = e } List
| . . | . . . . . | | . . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . . | | . . 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . . | | . .     =
| . . | . . . . . | | . .     ||sends(l;e)||
| . . | . . . . . | | . . 36. ||map(λe'.
| . . | . . . . . | | . .     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | | . .     =
| . . | . . . . . | | . .     ||sends(l;e)||
| . . | . . . . . | | . . 37. i1 : ℕ
| . . | . . . . . | | . . 38. i1<||map(λe'.
| . . | . . . . . | | . .     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | | . . 39. v : E
| . . | . . . . . | | . . 40. isrcv(v)
| . . | . . . . . | | . . 41. lnk(v) = l
| . . | . . . . . | | . . 42. sender(v) = e
| . . | . . . . . | | . . 43. index(v) = i1
| . . | . . . . . | | . . 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . . | | . .     ∃e':E.
| . . | . . . . . | | . .     isrcv(e')
| . . | . . . . . | | . .     & lnk(e') = l
| . . | . . . . . | | . .       & sender(e') = v
| . . | . . . . . | | . .       & index(e') = n
| . . | . . . . . | | . . 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . . | | . .     =
| . . | . . . . . | | . .     msg(lnk(v);tag(v);val(v))
| . . | . . . . . | | . . 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | | . . 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | | . . 48. z : Msg
```

```
| . . | . . . . . | | . . 49. z = sends(lnk(v);sender(v))[index(v)]
| . . | . . . . . | | . . 50. z ∈ (Msg on l)
| . . | . . . . . | | . . ⊢  z
| . . | . . . . . | | . .     ∈ IdLnk×(tg:Id×ma-valtype(da; rcv(l; tg)))
| . . | . . . . . | | . .    by DoSubsume
| . . | . . . . . | |
| . . | . . . . . | .\
| . . | . . . . . | . 1. i : Id
| . . | . . . . . | . 2. k : Knd
| . . | . . . . . | . 3. l : IdLnk
| . . | . . . . . | . 4. ds : x:Id fp-> Type
| . . | . . . . . | . 5. da : a:Knd fp-> Type
| . . | . . . . . | . 6. f :
| . . | . . . . . | . |  (tg:Id
| . . | . . . . . | . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . . | . |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . . | . 7. source(l) = i
| . . | . . . . . | . 8. D' : Dsys
| . . | . . . . . | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . . | . 10. w : World
| . . | . . . . . | . 11. p : FairFifo
| . . | . . . . . | . 12. PossibleWorld(D';w)
| . . | . . . . . | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | . 14. ∀e:E.
| . . | . . . . . | . |   loc(e) = i
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . 15. ∀e:E.
| . . | . . . . . | . |   isrcv(e)
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   lnk(e) = l
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | . 17. ∀e:E.
| . . | . . . . . | . |   loc(e) = i
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   kind(e) = k
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   sends(l;e)
| . . | . . . . . | . |   =
| . . | . . . . . | . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | . 19. ∀e:E.
| . . | . . . . . | . |   loc(e) = i
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . 20. ∀e:E.
| . . | . . . . . | . |   isrcv(e)
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   lnk(e) = l
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . 21. e : E
```

```
| . . | . . . . . | . 22. loc(e) = i
| . . | . . . . . | . 23. kind(e) = k
| . . | . . . . . | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | . 25. sends(l;e)
| . . | . . . . . | . |   =
| . . | . . . . . | . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | . 27. L :
| . . | . . . . . | . |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | . |   ∃e':E.
| . . | . . . . . | . |   isrcv(e')
| . . | . . . . . | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . | . 28. LL :
| . . | . . . . . | . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . . | . |                       | isrcv(e')
| . . | . . . . . | . |                         & lnk(e') = l
| . . | . . . . . | . |                           & sender(e') = e
| . . | . . . . . | . |                           & index(e') = n }
| . . | . . . . . | . 29. (λn.1of(L(n))) = LL
| . . | . . . . . | . 30. ∀e:E.
| . . | . . . . . | . |   isrcv(e)
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . | . |   =
| . . | . . . . . | . |   msg(lnk(e);tag(e);val(e))
| . . | . . . . . | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . | . |   ∃e':E.
| . . | . . . . . | . |   isrcv(e')
| . . | . . . . . | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . | . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . . | . |                         ×ma-valtype(da;
| . . | . . . . . | . |                                       rcv(l; tg)
| . . | . . . . . | . |                                       )))
| . . | . . . . . | . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . . | . |   ∈ {e':E
| . . | . . . . . | . |      | isrcv(e')
| . . | . . . . . | . |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . . | . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . . | . 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . . | . |   =
| . . | . . . . . | . |   ||sends(l;e)||
| . . | . . . . . | . 36. ||map(λe'.
| . . | . . . . . | . |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | . |   =
| . . | . . . . . | . |   ||sends(l;e)||
| . . | . . . . . | . 37. i1 : ℕ
| . . | . . . . . | . 38. i1<||map(λe'.
| . . | . . . . . | . |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | . 39. v : E
| . . | . . . . . | . 40. isrcv(v)
| . . | . . . . . | . 41. lnk(v) = l
| . . | . . . . . | . 42. sender(v) = e
| . . | . . . . . | . 43. index(v) = i1
| . . | . . . . . | . 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . . | . |   ∃e':E.
```

```
| . . | . . . . . | . |   isrcv(e')
| . . | . . . . . | . |   & lnk(e') = l & sender(e') = v & index(e') = n
| . . | . . . . . | . 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . . | . |   =
| . . | . . . . . | . |   msg(lnk(v);tag(v);val(v))
| . . | . . . . . | . 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | . 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | . 48. msg(lnk(v);tag(v);val(v)) = sends(l;e)[i1]
| . . | . . . . . | . ⊢  2of(msg(lnk(v);tag(v);val(v)))
| . . | . . . . . | . |  =
| . . | . . . . . | . |  2of(sends(l;e)[i1])
| . . | . . . . . | . | by Assert

| . . | . . . . . | . |    (val(v) ∈ ma-valtype(da; rcv(l; tag(v))))
| . . | . . . . . | . |    THEN
| . . | . . . . . | . |    Try
| . . | . . . . . | . |    (RevHypSubst -2 0 THEN Unfold `msg` 0
| . . | . . . . . | . |     THEN
| . . | . . . . . | . |     Reduce 0)
| . . | . . . . . | . |
| . . | . . . . . | . .....assertion .....
| . . | . . . . . | . 1. i : Id
| . . | . . . . . | . 2. k : Knd
| . . | . . . . . | . 3. l : IdLnk
| . . | . . . . . | . 4. ds : x:Id fp-> Type
| . . | . . . . . | . 5. da : a:Knd fp-> Type
| . . | . . . . . | . 6. f :
| . . | . . . . . | . |  (tg:Id
| . . | . . . . . | . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . . | . |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . . | . 7. source(l) = i
| . . | . . . . . | . 8. D' : Dsys
| . . | . . . . . | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . . | . 10. w : World
| . . | . . . . . | . 11. p : FairFifo
| . . | . . . . . | . 12. PossibleWorld(D';w)
| . . | . . . . . | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | . 14. ∀e:E.
| . . | . . . . . | . |   loc(e) = i
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . 15. ∀e:E.
| . . | . . . . . | . |   isrcv(e)
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   lnk(e) = l
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | . 17. ∀e:E.
| . . | . . . . . | . |   loc(e) = i
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   kind(e) = k
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   sends(l;e)
| . . | . . . . . | . |   =
```

```
| . . | . . . . . | . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | . 19. ∀e:E.
| . . | . . . . . | . |   loc(e) = i
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . 20. ∀e:E.
| . . | . . . . . | . |   isrcv(e)
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   lnk(e) = l
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . 21. e : E
| . . | . . . . . | . 22. loc(e) = i
| . . | . . . . . | . 23. kind(e) = k
| . . | . . . . . | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | . 25. sends(l;e)
| . . | . . . . . | . |   =
| . . | . . . . . | . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | . 27. L :
| . . | . . . . . | . |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | . |   ∃e':E.
| . . | . . . . . | . |   isrcv(e')
| . . | . . . . . | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . | . 28. LL :
| . . | . . . . . | . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . . | . |                      | isrcv(e')
| . . | . . . . . | . |                        & lnk(e') = l
| . . | . . . . . | . |                          & sender(e') = e
| . . | . . . . . | . |                          & index(e') = n }
| . . | . . . . . | . 29. (λn.1of(L(n))) = LL
| . . | . . . . . | . 30. ∀e:E.
| . . | . . . . . | . |   isrcv(e)
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . | . |   =
| . . | . . . . . | . |   msg(lnk(e);tag(e);val(e))
| . . | . . . . . | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . | . |   ∃e':E.
| . . | . . . . . | . |   isrcv(e')
| . . | . . . . . | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . | . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . . | . |                        ×ma-valtype(da;
| . . | . . . . . | . |                                    rcv(l; tg)
| . . | . . . . . | . |                                    )))
| . . | . . . . . | . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . . | . |   ∈ {e':E
| . . | . . . . . | . |      | isrcv(e')
| . . | . . . . . | . |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . . | . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . . | . 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . . | . |   =
| . . | . . . . . | . |   ||sends(l;e)||
| . . | . . . . . | . 36. ||map(λe'.
```

```
| . . | . . . . | . |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | . |   =
| . . | . . . . | . |   ||sends(l;e)||
| . . | . . . . | . 37. i1 : ℕ
| . . | . . . . | . 38. i1<||map(λe'.
| . . | . . . . | . |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | . 39. v : E
| . . | . . . . | . 40. isrcv(v)
| . . | . . . . | . 41. lnk(v) = l
| . . | . . . . | . 42. sender(v) = e
| . . | . . . . | . 43. index(v) = i1
| . . | . . . . | . 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | . |   ∃e':E.
| . . | . . . . | . |   isrcv(e')
| . . | . . . . | . |   & lnk(e') = l & sender(e') = v & index(e') = n
| . . | . . . . | . 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | . |   =
| . . | . . . . | . |   msg(lnk(v);tag(v);val(v))
| . . | . . . . | . 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | . 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | . 48. msg(lnk(v);tag(v);val(v)) = sends(l;e)[i1]
| . . | . . . . | . ⊢  val(v) ∈ ma-valtype(da; rcv(l; tag(v)))
| . . | . . . . | . | by DoSubsume
| . . | . . . . | . |
| . . | . . . . | . .....subtype_rel .....
| . . | . . . . | . 1. i : Id
| . . | . . . . | . 2. k : Knd
| . . | . . . . | . 3. l : IdLnk
| . . | . . . . | . 4. ds : x:Id fp-> Type
| . . | . . . . | . 5. da : a:Knd fp-> Type
| . . | . . . . | . 6. f :
| . . | . . . . | . |  (tg:Id
| . . | . . . . | . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | . |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . | . 7. source(l) = i
| . . | . . . . | . 8. D' : Dsys
| . . | . . . . | . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | . 10. w : World
| . . | . . . . | . 11. p : FairFifo
| . . | . . . . | . 12. PossibleWorld(D';w)
| . . | . . . . | . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | . 14. ∀e:E.
| . . | . . . . | . |   loc(e) = i
| . . | . . . . | . |   ⇒
| . . | . . . . | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | . 15. ∀e:E.
| . . | . . . . | . |   isrcv(e)
| . . | . . . . | . |   ⇒
| . . | . . . . | . |   lnk(e) = l
| . . | . . . . | . |   ⇒
| . . | . . . . | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . | . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . | . 17. ∀e:E.
| . . | . . . . | . |   loc(e) = i
```

```
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   kind(e) = k
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   sends(l;e)
| . . | . . . . . | . |   =
| . . | . . . . . | . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | . 19. ∀e:E.
| . . | . . . . . | . |   loc(e) = i
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . 20. ∀e:E.
| . . | . . . . . | . |   isrcv(e)
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   lnk(e) = l
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . 21. e : E
| . . | . . . . . | . 22. loc(e) = i
| . . | . . . . . | . 23. kind(e) = k
| . . | . . . . . | . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | . 25. sends(l;e)
| . . | . . . . . | . |   =
| . . | . . . . . | . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | . 27. L :
| . . | . . . . . | . |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | . |   ∃e':E.
| . . | . . . . . | . |   isrcv(e')
| . . | . . . . . | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . | . 28. LL :
| . . | . . . . . | . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . . | . |                       | isrcv(e')
| . . | . . . . . | . |                         & lnk(e') = l
| . . | . . . . . | . |                           & sender(e') = e
| . . | . . . . . | . |                           & index(e') = n }
| . . | . . . . . | . 29. (λn.1of(L(n))) = LL
| . . | . . . . . | . 30. ∀e:E.
| . . | . . . . . | . |   isrcv(e)
| . . | . . . . . | . |   ⇒
| . . | . . . . . | . |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . | . |   =
| . . | . . . . . | . |   msg(lnk(e);tag(e);val(e))
| . . | . . . . . | . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . | . |   ∃e':E.
| . . | . . . . . | . |   isrcv(e')
| . . | . . . . . | . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . | . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . . | . |                         ×ma-valtype(da;
| . . | . . . . . | . |                                     rcv(l; tg)
| . . | . . . . . | . |                                     )))
| . . | . . . . . | . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . . | . |   ∈ {e':E
| . . | . . . . . | . |      | isrcv(e')
| . . | . . . . . | . |        & lnk(e') = l & sender(e') = e } List
```

```
| . . | . . . . | . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . | . 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . | . |   =
| . . | . . . . | . |   ||sends(l;e)||
| . . | . . . . | . 36. ||map(λe'.
| . . | . . . . | . |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | . |   =
| . . | . . . . | . |   ||sends(l;e)||
| . . | . . . . | . 37. i1 : ℕ
| . . | . . . . | . 38. i1<||map(λe'.
| . . | . . . . | . |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . | . 39. v : E
| . . | . . . . | . 40. isrcv(v)
| . . | . . . . | . 41. lnk(v) = l
| . . | . . . . | . 42. sender(v) = e
| . . | . . . . | . 43. index(v) = i1
| . . | . . . . | . 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . | . |   ∃e':E.
| . . | . . . . | . |   isrcv(e')
| . . | . . . . | . |   & lnk(e') = l & sender(e') = v & index(e') = n
| . . | . . . . | . 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . | . |   =
| . . | . . . . | . |   msg(lnk(v);tag(v);val(v))
| . . | . . . . | . 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | . 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . | . 48. msg(lnk(v);tag(v);val(v)) = sends(l;e)[i1]
| . . | . . . . | . ⊢  valtype(v) ⊆r ma-valtype(da; rcv(l; tag(v)))
| . . | . . . . | . | by Subst' (rcv(l; tag(v)) = kind(v)) 0
| . . | . . . . | . |
| . . | . . . . | . |\
| . . | . . . . | . | 1. i : Id
| . . | . . . . | . | 2. k : Knd
| . . | . . . . | . | 3. l : IdLnk
| . . | . . . . | . | 4. ds : x:Id fp-> Type
| . . | . . . . | . | 5. da : a:Knd fp-> Type
| . . | . . . . | . | 6. f :
| . . | . . . . | . |    (tg:Id
| . . | . . . . | . |     ×State(ds)→ma-valtype(da; k)→
| . . | . . . . | . |     (da(rcv(l; tg))?Void List)) List
| . . | . . . . | . | 7. source(l) = i
| . . | . . . . | . | 8. D' : Dsys
| . . | . . . . | . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . | . | 10. w : World
| . . | . . . . | . | 11. p : FairFifo
| . . | . . . . | . | 12. PossibleWorld(D';w)
| . . | . . . . | . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . | . | 14. ∀e:E.
| . . | . . . . | . |     loc(e) = i
| . . | . . . . | . |     ⇒
| . . | . . . . | . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . | . | 15. ∀e:E.
| . . | . . . . | . |     isrcv(e)
| . . | . . . . | . |     ⇒
| . . | . . . . | . |     lnk(e) = l
| . . | . . . . | . |     ⇒
```

```
| . . | . . . . . | . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | . |     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | . | 17. ∀e:E.
| . . | . . . . . | . |     loc(e) = i
| . . | . . . . . | . |     ⇒
| . . | . . . . . | . |     kind(e) = k
| . . | . . . . . | . |     ⇒
| . . | . . . . . | . |     sends(l;e)
| . . | . . . . . | . |     =
| . . | . . . . . | . |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | . | 19. ∀e:E.
| . . | . . . . . | . |     loc(e) = i
| . . | . . . . . | . |     ⇒
| . . | . . . . . | . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . | 20. ∀e:E.
| . . | . . . . . | . |     isrcv(e)
| . . | . . . . . | . |     ⇒
| . . | . . . . . | . |     lnk(e) = l
| . . | . . . . . | . |     ⇒
| . . | . . . . . | . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . | 21. e : E
| . . | . . . . . | . | 22. loc(e) = i
| . . | . . . . . | . | 23. kind(e) = k
| . . | . . . . . | . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | . | 25. sends(l;e)
| . . | . . . . . | . |     =
| . . | . . . . . | . |     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | . | 27. L :
| . . | . . . . . | . |     ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | . |     ∃e':E.
| . . | . . . . . | . |     isrcv(e')
| . . | . . . . . | . |     & lnk(e') = l
| . . | . . . . . | . |       & sender(e') = e
| . . | . . . . . | . |       & index(e') = n

| . . | . . . . . | . | 28. LL :
| . . | . . . . . | . |     n:ℕ||sends(l;e)||→{e':E
| . . | . . . . . | . |                          | isrcv(e')
| . . | . . . . . | . |                            & lnk(e') = l
| . . | . . . . . | . |                              & sender(e') = e
| . . | . . . . . | . |                              & index(e') = n }
| . . | . . . . . | . | 29. (λn.1of(L(n))) = LL
| . . | . . . . . | . | 30. ∀e:E.
| . . | . . . . . | . |     isrcv(e)
| . . | . . . . . | . |     ⇒
| . . | . . . . . | . |     sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . | . |     =
| . . | . . . . . | . |     msg(lnk(e);tag(e);val(e))
| . . | . . . . . | . | 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . | . |     ∃e':E.
| . . | . . . . . | . |     isrcv(e')
| . . | . . . . . | . |     & lnk(e') = l
| . . | . . . . . | . |       & sender(e') = e
```

```
| . . | . . . . . | . . |       & index(e') = n
| . . | . . . . . | . . | 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . . | . . |                             ×ma-valtype(da;
| . . | . . . . . | . . |                                         rcv(l; tg)
| . . | . . . . . | . . |                                         )))
| . . | . . . . . | . . | 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . . | . . |     ∈ {e':E
| . . | . . . . . | . . |        | isrcv(e')
| . . | . . . . . | . . |          & lnk(e') = l & sender(e') = e } List
| . . | . . . . . | . . | 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . . | . . | 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . . | . . |     =
| . . | . . . . . | . . |     ||sends(l;e)||
| . . | . . . . . | . . | 36. ||map(λe'.
| . . | . . . . . | . . |     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | . . |     =
| . . | . . . . . | . . |     ||sends(l;e)||
| . . | . . . . . | . . | 37. i1 : ℕ
| . . | . . . . . | . . | 38. i1<||map(λe'.
| . . | . . . . . | . . |     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | . . | 39. v : E
| . . | . . . . . | . . | 40. isrcv(v)
| . . | . . . . . | . . | 41. lnk(v) = l
| . . | . . . . . | . . | 42. sender(v) = e
| . . | . . . . . | . . | 43. index(v) = i1
| . . | . . . . . | . . | 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . . | . . |     ∃e':E.
| . . | . . . . . | . . |     isrcv(e')
| . . | . . . . . | . . |     & lnk(e') = l
| . . | . . . . . | . . |       & sender(e') = v
| . . | . . . . . | . . |       & index(e') = n
| . . | . . . . . | . . | 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . . | . . |     =
| . . | . . . . . | . . |     msg(lnk(v);tag(v);val(v))
| . . | . . . . . | . . | 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | . . | 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | . . | 48. msg(lnk(v);tag(v);val(v)) = sends(l;e)[i1]
| . . | . . . . . | . . | ⊢  rcv(l; tag(v)) = kind(v)
| . . | . . . . . | . . |   by Subst' (l = lnk(v)) 0
| . . | . . . . . | . . |      THEN
| . . | . . . . . | . . |      AssertBY isrcv(v) Trivial
| . . | . . . . . | . . |      THEN
| . . | . . . . . | . . |      MoveToConcl -1
| . . | . . . . . | . . |      THEN
| . . | . . . . . | . . |      Unfolds [`es-isrcv`;`es-tag`;`es-lnk`] 0
| . . | . . . . . | . . |      THEN
| . . | . . . . . | . . |      GenConclAtAddr [2;3]
| . . | . . . . . | . . |      THEN
| . . | . . . . . | . . |      Thin -1
| . . | . . . . . | . . |      THEN
| . . | . . . . . | . . |      Unfolds [`isrcv`;`lnk`;`tagof`] 0
| . . | . . . . . | . . |      THEN
| . . | . . . . . | . . |      Analyze -1
| . . | . . . . . | . . |      THEN
| . . | . . . . . | . . |      Analyze -1
```

```
| . . | . . . . . | . |      THEN
| . . | . . . . . | . |      Reduce 0
| . . | . . . . . | . |      THEN
| . . | . . . . . | . |      Fold ‘rcv‘ 0
| . . | . . . . . | . |      THEN
| . . | . . . . . | . |      Analyze
| . . | . . . . . | . |
| . . | . . . . . | . .\
| . . | . . . . . | . . 1. i : Id
| . . | . . . . . | . . 2. k : Knd
| . . | . . . . . | . . 3. l : IdLnk
| . . | . . . . . | . . 4. ds : x:Id fp-> Type
| . . | . . . . . | . . 5. da : a:Knd fp-> Type
| . . | . . . . . | . . 6. f :
| . . | . . . . . | . .    (tg:Id
| . . | . . . . . | . .     ×State(ds)→ma-valtype(da; k)→
| . . | . . . . . | . .     (da(rcv(l; tg))?Void List)) List
| . . | . . . . . | . . 7. source(l) = i
| . . | . . . . . | . . 8. D’ : Dsys
| . . | . . . . . | . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D’
| . . | . . . . . | . . 10. w : World
| . . | . . . . . | . . 11. p : FairFifo
| . . | . . . . . | . . 12. PossibleWorld(D’;w)
| . . | . . . . . | . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | . . 14. ∀e:E.
| . . | . . . . . | . .     loc(e) = i
| . . | . . . . . | . .     ⇒
| . . | . . . . . | . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . . 15. ∀e:E.
| . . | . . . . . | . .     isrcv(e)
| . . | . . . . . | . .     ⇒
| . . | . . . . . | . .     lnk(e) = l
| . . | . . . . . | . .     ⇒
| . . | . . . . . | . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . | . .     ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . | . . 17. ∀e:E.
| . . | . . . . . | . .     loc(e) = i
| . . | . . . . . | . .     ⇒
| . . | . . . . . | . .     kind(e) = k
| . . | . . . . . | . .     ⇒
| . . | . . . . . | . .     sends(l;e)
| . . | . . . . . | . .     =
| . . | . . . . . | . .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . | . . 19. ∀e:E.
| . . | . . . . . | . .     loc(e) = i
| . . | . . . . . | . .     ⇒
| . . | . . . . . | . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . | . . 20. ∀e:E.
| . . | . . . . . | . .     isrcv(e)
| . . | . . . . . | . .     ⇒
| . . | . . . . . | . .     lnk(e) = l
| . . | . . . . . | . .     ⇒
| . . | . . . . . | . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
```

```
| . . | . . . . . | . . 21. e : E
| . . | . . . . . | . . 22. loc(e) = i
| . . | . . . . . | . . 23. kind(e) = k
| . . | . . . . . | . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | . . 25. sends(l;e)
| . . | . . . . . | . .     =
| . . | . . . . . | . .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . | . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . | . . 27. L :
| . . | . . . . . | . .     ∀n:ℕ||sends(l;e)||.
| . . | . . . . . | . .     ∃e':E.
| . . | . . . . . | . .     isrcv(e')
| . . | . . . . . | . .     & lnk(e') = l
| . . | . . . . . | . .       & sender(e') = e
| . . | . . . . . | . .       & index(e') = n
| . . | . . . . . | . . 28. LL :
| . . | . . . . . | . .     n:ℕ||sends(l;e)||→{e':E
| . . | . . . . . | . .                         | isrcv(e')
| . . | . . . . . | . .                           & lnk(e') = l
| . . | . . . . . | . .                             & sender(e') = e
| . . | . . . . . | . .                             & index(e') = n }
| . . | . . . . . | . . 29. (λn.1of(L(n))) = LL
| . . | . . . . . | . . 30. ∀e:E.
| . . | . . . . . | . .     isrcv(e)
| . . | . . . . . | . .     ⇒
| . . | . . . . . | . .     sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . | . .     =
| . . | . . . . . | . .     msg(lnk(e);tag(e);val(e))
| . . | . . . . . | . . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . | . .     ∃e':E.
| . . | . . . . . | . .     isrcv(e')
| . . | . . . . . | . .     & lnk(e') = l
| . . | . . . . . | . .       & sender(e') = e
| . . | . . . . . | . .       & index(e') = n
| . . | . . . . . | . . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . . | . .                           ×ma-valtype(da;
| . . | . . . . . | . .                                         rcv(l; tg)
| . . | . . . . . | . .                                         )))
| . . | . . . . . | . . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . . | . .     ∈ {e':E
| . . | . . . . . | . .        | isrcv(e')
| . . | . . . . . | . .          & lnk(e') = l & sender(e') = e } List
| . . | . . . . . | . . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . . | . . 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . . | . .     =
| . . | . . . . . | . .     ||sends(l;e)||
| . . | . . . . . | . . 36. ||map(λe'.
| . . | . . . . . | . .     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | . .     =
| . . | . . . . . | . .     ||sends(l;e)||
| . . | . . . . . | . . 37. i1 : ℕ
| . . | . . . . . | . . 38. i1<||map(λe'.
| . . | . . . . . | . .     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . | . . 39. v : E
| . . | . . . . . | . . 40. isrcv(v)
```

```
| . . | . . . . . | . . 41. lnk(v) = l
| . . | . . . . . | . . 42. sender(v) = e
| . . | . . . . . | . . 43. index(v) = i1
| . . | . . . . . | . . 44. ∀l:IdLnk, n:ℕ||sends(l;v)||.
| . . | . . . . . | . .     ∃e':E.
| . . | . . . . . | . .     isrcv(e')
| . . | . . . . . | . .     & lnk(e') = l
| . . | . . . . . | . .       & sender(e') = v
| . . | . . . . . | . .       & index(e') = n
| . . | . . . . . | . . 45. sends(lnk(v);sender(v))[index(v)]
| . . | . . . . . | . .     =
| . . | . . . . . | . .     msg(lnk(v);tag(v);val(v))
| . . | . . . . . | . . 46. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | . . 47. valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | . . 48. msg(lnk(v);tag(v);val(v)) = sends(l;e)[i1]
| . . | . . . . . | . . ⊢  valtype(v) ⊆r ma-valtype(da; kind(v))
| . . | . . . . . | . .   by BackThruSomeHyp

| . . | . . . . . |
| . . | . . . . . .\
| . . | . . . . . .....wf .....
| . . | . . . . . 1. i : Id
| . . | . . . . . 2. k : Knd
| . . | . . . . . 3. l : IdLnk
| . . | . . . . . 4. ds : x:Id fp-> Type
| . . | . . . . . 5. da : a:Knd fp-> Type
| . . | . . . . . 6. f :
| . . | . . . . . |  (tg:Id
| . . | . . . . . |   ×State(ds)→ma-valtype(da; k)→
| . . | . . . . . |   (da(rcv(l; tg))?Void List)) List
| . . | . . . . . 7. source(l) = i
| . . | . . . . . 8. D' : Dsys
| . . | . . . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . . 10. w : World
| . . | . . . . . 11. p : FairFifo
| . . | . . . . . 12. PossibleWorld(D';w)
| . . | . . . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . 14. ∀e:E.
| . . | . . . . . |   loc(e) = i
| . . | . . . . . |   ⇒
| . . | . . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . 15. ∀e:E.
| . . | . . . . . |   isrcv(e)
| . . | . . . . . |   ⇒
| . . | . . . . . |   lnk(e) = l
| . . | . . . . . |   ⇒
| . . | . . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . 17. ∀e:E.
| . . | . . . . . |   loc(e) = i
| . . | . . . . . |   ⇒
| . . | . . . . . |   kind(e) = k
| . . | . . . . . |   ⇒
| . . | . . . . . |   sends(l;e)
| . . | . . . . . |   =
```

```
| . . | . . . . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . 19. ∀e:E.
| . . | . . . . . |   loc(e) = i
| . . | . . . . . |   ⇒
| . . | . . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . 20. ∀e:E.
| . . | . . . . . |   isrcv(e)
| . . | . . . . . |   ⇒
| . . | . . . . . |   lnk(e) = l
| . . | . . . . . |   ⇒
| . . | . . . . . |   (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . 21. e : E
| . . | . . . . . 22. loc(e) = i
| . . | . . . . . 23. kind(e) = k
| . . | . . . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . 25. sends(l;e)
| . . | . . . . . |   =
| . . | . . . . . |   tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . 27. L :
| . . | . . . . . |   ∀n:ℕ||sends(l;e)||.
| . . | . . . . . |   ∃e':E.
| . . | . . . . . |   isrcv(e')
| . . | . . . . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . 28. LL :
| . . | . . . . . |   n:ℕ||sends(l;e)||→{e':E
| . . | . . . . . |                      | isrcv(e')
| . . | . . . . . |                        & lnk(e') = l
| . . | . . . . . |                          & sender(e') = e
| . . | . . . . . |                          & index(e') = n }
| . . | . . . . . 29. (λn.1of(L(n))) = LL
| . . | . . . . . 30. ∀e:E.
| . . | . . . . . |   isrcv(e)
| . . | . . . . . |   ⇒
| . . | . . . . . |   sends(lnk(e);sender(e))[index(e)]
| . . | . . . . . |   =
| . . | . . . . . |   msg(lnk(e);tag(e);val(e))
| . . | . . . . . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . . |   ∃e':E.
| . . | . . . . . |   isrcv(e')
| . . | . . . . . |   & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . . |                        ×ma-valtype(da; rcv(l; tg)
| . . | . . . . . |                                    )))
| . . | . . . . . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . . |   ∈ {e':E
| . . | . . . . . |      | isrcv(e')
| . . | . . . . . |        & lnk(e') = l & sender(e') = e } List
| . . | . . . . . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . . 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . . |   =
| . . | . . . . . |   ||sends(l;e)||
| . . | . . . . . 36. ||map(λe'.
| . . | . . . . . |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
```

```
| . . | . . . . . . |   =
| . . | . . . . . . |   ||sends(l;e)||
| . . | . . . . . . 37. i1 : ℕ
| . . | . . . . . . 38. i1<||map(λe'.
| . . | . . . . . . |   <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . . 39. v : E
| . . | . . . . . . 40. isrcv(v)
| . . | . . . . . . 41. lnk(v) = l
| . . | . . . . . . 42. sender(v) = e
| . . | . . . . . . 43. index(v) = i1
| . . | . . . . . . 44. z : tg:Id×ma-valtype(da; rcv(l; tg))
| . . | . . . . . . ⊢  (z = 2of(sends(l;e)[i1])) ∈ Prop  by Auto
| . . | . . . . . . |
| . . | . . . . . . .....subterm .....
| . . | . . . . . . 1. i : Id
| . . | . . . . . . 2. k : Knd
| . . | . . . . . . 3. l : IdLnk
| . . | . . . . . . 4. ds : x:Id fp-> Type
| . . | . . . . . . 5. da : a:Knd fp-> Type
| . . | . . . . . . 6. f :
| . . | . . . . .      (tg:Id
| . . | . . . . .       ×State(ds)→ma-valtype(da; k)→
| . . | . . . . .       (da(rcv(l; tg))?Void List)) List
| . . | . . . . . . 7. source(l) = i
| . . | . . . . . . 8. D' : Dsys
| . . | . . . . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . | . . . . . . 10. w : World
| . . | . . . . . . 11. p : FairFifo
| . . | . . . . . . 12. PossibleWorld(D';w)
| . . | . . . . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . . 14. ∀e:E.
| . . | . . . . .       loc(e) = i
| . . | . . . . .       ⇒
| . . | . . . . .       (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . . 15. ∀e:E.
| . . | . . . . .       isrcv(e)
| . . | . . . . .       ⇒
| . . | . . . . .       lnk(e) = l
| . . | . . . . .       ⇒
| . . | . . . . .       (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . | . . . . .       ((λl,tg. da(rcv(l; tg))?Top))
| . . | . . . . . . 17. ∀e:E.
| . . | . . . . .       loc(e) = i
| . . | . . . . .       ⇒
| . . | . . . . .       kind(e) = k
| . . | . . . . .       ⇒
| . . | . . . . .       sends(l;e)
| . . | . . . . .       =
| . . | . . . . .       tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . | . . . . . . 19. ∀e:E.
| . . | . . . . .       loc(e) = i
| . . | . . . . .       ⇒
| . . | . . . . .       (valtype(e) ⊆r ma-valtype(da; kind(e)))
```

```
| . . | . . . . . 20. ∀e:E.
| . . | . . . . .     isrcv(e)
| . . | . . . . .     ⇒
| . . | . . . . .     lnk(e) = l
| . . | . . . . .     ⇒
| . . | . . . . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . | . . . . . 21. e : E
| . . | . . . . . 22. loc(e) = i
| . . | . . . . . 23. kind(e) = k
| . . | . . . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . 25. sends(l;e)
| . . | . . . . .     =
| . . | . . . . .     tagged-messages(l;λz.(z when e);val(e);f)
| . . | . . . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . | . . . . . 27. L :
| . . | . . . . .     ∀n:ℕ||sends(l;e)||.
| . . | . . . . .     ∃e':E.
| . . | . . . . .     isrcv(e')
| . . | . . . . .     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . 28. LL :
| . . | . . . . .     n:ℕ||sends(l;e)||→{e':E
| . . | . . . . .                          | isrcv(e')
| . . | . . . . .                            & lnk(e') = l
| . . | . . . . .                              & sender(e') = e
| . . | . . . . .                              & index(e') = n }
| . . | . . . . . 29. (λn.1of(L(n))) = LL
| . . | . . . . . 30. ∀e:E.
| . . | . . . . .     isrcv(e)
| . . | . . . . .     ⇒
| . . | . . . . .     sends(lnk(e);sender(e))[index(e)]
| . . | . . . . .     =
| . . | . . . . .     msg(lnk(e);tag(e);val(e))
| . . | . . . . . 31. ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
| . . | . . . . .     ∃e':E.
| . . | . . . . .     isrcv(e')
| . . | . . . . .     & lnk(e') = l & sender(e') = e & index(e') = n
| . . | . . . . . 32. (Msg on l) ⊆r (IdLnk×(tg:Id
| . . | . . . . .                            ×ma-valtype(da; rcv(l; tg)
| . . | . . . . .                                           )))
| . . | . . . . . 33. map(LL;upto(||sends(l;e)||))
| . . | . . . . .     ∈ {e':E
| . . | . . . . .        | isrcv(e')
| . . | . . . . .          & lnk(e') = l & sender(e') = e } List
| . . | . . . . . 34. ||upto(||sends(l;e)||)|| = ||sends(l;e)||
| . . | . . . . . 35. ||map(LL;upto(||sends(l;e)||))||
| . . | . . . . .     =
| . . | . . . . .     ||sends(l;e)||
| . . | . . . . . 36. ||map(λe'.
| . . | . . . . .     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . .     =
| . . | . . . . .     ||sends(l;e)||
| . . | . . . . . 37. i1 : ℕ
| . . | . . . . . 38. i1<||map(λe'.
| . . | . . . . .     <tag(e'),val(e')>;map(LL;upto(||...||)))||
| . . | . . . . . 39. v : E
```

```
| . . | . . . . . 40. isrcv(v)
| . . | . . . . . 41. lnk(v) = l
| . . | . . . . . 42. sender(v) = e
| . . | . . . . . 43. index(v) = i1
| . . | . . . . . 44. tg:Id×ma-valtype(da; rcv(l; tg))
| . . | . . . . . ⊢  2of(sends(l;e)[i1])
| . . | . . . . .     ∈ tg:Id×ma-valtype(da; rcv(l; tg))
| . . | . . . . .   by GenConcl (sends(l;e)[i1] = X)
| . . |
| . . .\
| . . . .....wf .....
| . . . 1. i : Id
| . . . 2. k : Knd
| . . . 3. l : IdLnk
| . . . 4. ds : x:Id fp-> Type
| . . . 5. da : a:Knd fp-> Type
| . . . 6. f :
| . . . |  (tg:Id
| . . . |   ×State(ds)→ma-valtype(da; k)→
| . . . |   (da(rcv(l; tg))?Void List)) List
| . . . 7. source(l) = i
| . . . 8. D' : Dsys
| . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . . 10. w : World
| . . . 11. p : FairFifo
| . . . 12. PossibleWorld(D';w)
| . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . 14. ∀e:E.
| . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . 15. ∀e:E.
| . . . |   isrcv(e)
| . . . |   ⇒
| . . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . . 17. ∀e:E.
| . . . |   loc(e) = i
| . . . |   ⇒
| . . . |   kind(e) = k
| . . . |   ⇒
| . . . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . 19. ∀e:E.
| . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . 20. ∀e:E.
| . . . |   isrcv(e)
| . . . |   ⇒
| . . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . 21. e : E
| . . . 22. loc(e) = i
| . . . 23. kind(e) = k
| . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . 27. L :
```

```
| . . . |   ∀n:ℕ||sends(l;e)||.
| . . . |   ∃e':E.
| . . . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . . 28. LL :
| . . . |   n:ℕ||sends(l;e)||→{e':E
| . . . |                     | isrcv(e')
| . . . |                       & lnk(e') = l
| . . . |                         & sender(e') = e
| . . . |                         & index(e') = n }
| . . . 29. (λn.1of(L(n))) = LL
| . . . 30. z : (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . . ⊢  (map(λe'.<tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))
| . . . |   =
| . . . |   z)
| . . . |  ∈ Prop  by Auto
| . . . |
| . . . .....subterm .....
| . . . 1. i : Id
| . . . 2. k : Knd
| . . . 3. l : IdLnk
| . . . 4. ds : x:Id fp-> Type
| . . . 5. da : a:Knd fp-> Type
| . . . 6. f :
| . . . |  (tg:Id
| . . . |   ×State(ds)→ma-valtype(da; k)→
| . . . |   (da(rcv(l; tg))?Void List)) List
| . . . 7. source(l) = i
| . . . 8. D' : Dsys
| . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . . 10. w : World
| . . . 11. p : FairFifo
| . . . 12. PossibleWorld(D';w)
| . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . 14. ∀e:E.
| . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . 15. ∀e:E.
| . . . |   isrcv(e)
| . . . |   ⇒
| . . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . . 17. ∀e:E.
| . . . |   loc(e) = i
| . . . |   ⇒
| . . . |   kind(e) = k
| . . . |   ⇒
| . . . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . 19. ∀e:E.
| . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . 20. ∀e:E.
| . . . |   isrcv(e)
| . . . |   ⇒
| . . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . 21. e : E
```

```
| . . . 22. loc(e) = i
| . . . 23. kind(e) = k
| . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . 27. L :
| . . . |   ∀n:ℕ||sends(l;e)||.
| . . . |   ∃e':E.
| . . . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . . 28. LL :
| . . . |   n:ℕ||sends(l;e)||→{e':E
| . . . |                       | isrcv(e')
| . . . |                         & lnk(e') = l
| . . . |                           & sender(e') = e
| . . . |                           & index(e') = n }
| . . . 29. (λn.1of(L(n))) = LL
| . . . 30. (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . . ⊢  map(λe'.<tag(e'),val(e')>;map(LL;upto(||sends(l;e)||)))
| . . . |  ∈ (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . . | by GenConcl (map(LL;upto(||sends(l;e)||)) = X)
| . . . |
| . . . |\
| . . . | .....wf .....
| . . . | 1. i : Id
| . . . | 2. k : Knd
| . . . | 3. l : IdLnk
| . . . | 4. ds : x:Id fp-> Type
| . . . | 5. da : a:Knd fp-> Type
| . . . | 6. f :
| . . . | |  (tg:Id
| . . . | |   ×State(ds)→ma-valtype(da; k)→
| . . . | |   (da(rcv(l; tg))?Void List)) List
| . . . | 7. source(l) = i
| . . . | 8. D' : Dsys
| . . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . . | 10. w : World
| . . . | 11. p : FairFifo
| . . . | 12. PossibleWorld(D';w)
| . . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . | 14. ∀e:E.
| . . . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . | 15. ∀e:E.
| . . . | |   isrcv(e)
| . . . | |   ⇒
| . . . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . . | |   ((λl,tg. da(rcv(l; tg))?Top))
| . . . | 17. ∀e:E.
| . . . | |   loc(e) = i
| . . . | |   ⇒
| . . . | |   kind(e) = k
| . . . | |   ⇒
| . . . | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . | 19. ∀e:E.
```

```
| . . . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . | 20. ∀e:E.
| . . . | |   isrcv(e)
| . . . | |   ⇒
| . . . | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . | 21. e : E
| . . . | 22. loc(e) = i
| . . . | 23. kind(e) = k
| . . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . | 27. L :
| . . . | |   ∀n:ℕ||sends(l;e)||.
| . . . | |   ∃e':E.
| . . . | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . . | 28. LL :
| . . . | |   n:ℕ||sends(l;e)||→{e':E
| . . . | |                      | isrcv(e')
| . . . | |                        & lnk(e') = l
| . . . | |                          & sender(e') = e
| . . . | |                          & index(e') = n }
| . . . | 29. (λn.1of(L(n))) = LL
| . . . | 30. (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . . | ⊢  map(LL;upto(||sends(l;e)||))
| . . . | |  ∈ {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . . | | by Auto
| . . . | |
| . . . | .....subterm .....
| . . . | 1. i : Id
| . . . | 2. k : Knd
| . . . | 3. l : IdLnk
| . . . | 4. ds : x:Id fp-> Type
| . . . | 5. da : a:Knd fp-> Type
| . . . | 6. f :
| . . . |    (tg:Id
| . . . |     ×State(ds)→ma-valtype(da; k)→
| . . . |     (da(rcv(l; tg))?Void List)) List
| . . . | 7. source(l) = i
| . . . | 8. D' : Dsys
| . . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . . | 10. w : World
| . . . | 11. p : FairFifo
| . . . | 12. PossibleWorld(D';w)
| . . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . | 14. ∀e:E.
| . . . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . | 15. ∀e:E.
| . . . |     isrcv(e)
| . . . |     ⇒
| . . . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . . |     ((λl,tg. da(rcv(l; tg))?Top))
| . . . | 17. ∀e:E.
| . . . |     loc(e) = i
| . . . |     ⇒
```

```
| . . . |     kind(e) = k
| . . . |     ⇒
| . . . |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . | 19. ∀e:E.
| . . . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . | 20. ∀e:E.
| . . . |     isrcv(e)
| . . . |     ⇒
| . . . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . | 21. e : E
| . . . | 22. loc(e) = i
| . . . | 23. kind(e) = k
| . . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . | 27. L :
| . . . |     ∀n:ℕ||sends(l;e)||.
| . . . |     ∃e':E.
| . . . |     isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . . | 28. LL :
| . . . |     n:ℕ||sends(l;e)||→{e':E
| . . . |                          | isrcv(e')
| . . . |                            & lnk(e') = l
| . . . |                              & sender(e') = e
| . . . |                              & index(e') = n }
| . . . | 29. (λn.1of(L(n))) = LL
| . . . | 30. (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . . | ⊢  LL
| . . . |    ∈ ℕ||sends(l;e)||→{e':E
| . . . |                         | isrcv(e')
| . . . |                           & lnk(e') = l & sender(e') = e }
| . . . |   by Ext2 THEN DoSubsume THEN Analyze 0 THEN Analyze -1
| . . . |      THEN
| . . . |      Analyze
| . . . |      THEN
| . . . |      ExRepD
| . . . |
| . . . |\
| . . . | .....wf .....
| . . . | 1. i : Id
| . . . | 2. k : Knd
| . . . | 3. l : IdLnk
| . . . | 4. ds : x:Id fp-> Type
| . . . | 5. da : a:Knd fp-> Type
| . . . | 6. f :
| . . . |    (tg:Id
| . . . |     ×State(ds)→ma-valtype(da; k)→
| . . . |     (da(rcv(l; tg))?Void List)) List
| . . . | 7. source(l) = i
| . . . | 8. D' : Dsys
| . . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . . | 10. w : World
| . . . | 11. p : FairFifo
| . . . | 12. PossibleWorld(D';w)
```

```
| . . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . | 14. ∀e:E.
| . . . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . | 15. ∀e:E.
| . . . |     isrcv(e)
| . . . |     ⇒
| . . . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . . |     ((λl,tg. da(rcv(l; tg))?Top))
| . . . | 17. ∀e:E.
| . . . |     loc(e) = i
| . . . |     ⇒
| . . . |     kind(e) = k
| . . . |     ⇒
| . . . |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . | 19. ∀e:E.
| . . . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . | 20. ∀e:E.
| . . . |     isrcv(e)
| . . . |     ⇒
| . . . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . | 21. e : E
| . . . | 22. loc(e) = i
| . . . | 23. kind(e) = k
| . . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . | 27. L :
| . . . |     ∀n:ℕ||sends(l;e)||.
| . . . |     ∃e':E.
| . . . |     isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . . | 28. LL :
| . . . |     n:ℕ||sends(l;e)||→{e':E
| . . . |                       | isrcv(e')
| . . . |                         & lnk(e') = l
| . . . |                           & sender(e') = e
| . . . |                           & index(e') = n }
| . . . | 29. (λn.1of(L(n))) = LL
| . . . | 30. (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . . | ⊢  {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . . |    ∈ Type  by Auto
| . . . |
| . . . .\
| . . . . 1. i : Id
| . . . . 2. k : Knd
| . . . . 3. l : IdLnk
| . . . . 4. ds : x:Id fp-> Type
| . . . . 5. da : a:Knd fp-> Type
| . . . . 6. f :
| . . . . |  (tg:Id
| . . . . |   ×State(ds)→ma-valtype(da; k)→
| . . . . |   (da(rcv(l; tg))?Void List)) List
| . . . . 7. source(l) = i
| . . . . 8. D' : Dsys
```

```
| . . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . . . 10. w : World
| . . . . 11. p : FairFifo
| . . . . 12. PossibleWorld(D';w)
| . . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . . 14. ∀e:E.
| . . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . 15. ∀e:E.
| . . . . |   isrcv(e)
| . . . . |   ⇒
| . . . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . . . 17. ∀e:E.
| . . . . |   loc(e) = i
| . . . . |   ⇒
| . . . . |   kind(e) = k
| . . . . |   ⇒
| . . . . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . . 19. ∀e:E.
| . . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . 20. ∀e:E.
| . . . . |   isrcv(e)
| . . . . |   ⇒
| . . . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . 21. e : E
| . . . . 22. loc(e) = i
| . . . . 23. kind(e) = k
| . . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . . 27. L :
| . . . . |   ∀n:ℕ||sends(l;e)||.
| . . . . |   ∃e':E.
| . . . . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . . . 28. LL :
| . . . . |   n:ℕ||sends(l;e)||→{e':E
| . . . . |                        | isrcv(e')
| . . . . |                          & lnk(e') = l
| . . . . |                            & sender(e') = e
| . . . . |                            & index(e') = n }
| . . . . 29. (λn.1of(L(n))) = LL
| . . . . 30. (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . . . 31. X :
| . . . . |   {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . . . 32. map(LL;upto(||sends(l;e)||)) = X
| . . . . ⊢  map(λe'.<tag(e'),val(e')>;X)
| . . . . |  ∈ (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . . . | by Auto THEN Analyze -1 THEN ExRepD THEN Unhide
| . . . . |    THEN
| . . . . |    ExRepD
| . . . . |    THEN
| . . . . |    DoSubsume
| . . . . |
```

```
| . . . . .....subtype_rel .....
| . . . . 1. i : Id
| . . . . 2. k : Knd
| . . . . 3. l : IdLnk
| . . . . 4. ds : x:Id fp-> Type
| . . . . 5. da : a:Knd fp-> Type
| . . . . 6. f :
| . . . . |  (tg:Id
| . . . . |   ×State(ds)→ma-valtype(da; k)→
| . . . . |   (da(rcv(l; tg))?Void List)) List
| . . . . 7. source(l) = i
| . . . . 8. D' : Dsys
| . . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . . . 10. w : World
| . . . . 11. p : FairFifo
| . . . . 12. PossibleWorld(D';w)
| . . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . . 14. ∀e:E.
| . . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . 15. ∀e:E.
| . . . . |   isrcv(e)
| . . . . |   ⇒
| . . . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . . . |   ((λl,tg. da(rcv(l; tg))?Top))
| . . . . 17. ∀e:E.
| . . . . |   loc(e) = i
| . . . . |   ⇒
| . . . . |   kind(e) = k
| . . . . |   ⇒
| . . . . |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . . 19. ∀e:E.
| . . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . 20. ∀e:E.
| . . . . |   isrcv(e)
| . . . . |   ⇒
| . . . . |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . 21. e : E
| . . . . 22. loc(e) = i
| . . . . 23. kind(e) = k
| . . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . . 27. L :
| . . . . |   ∀n:ℕ||sends(l;e)||.
| . . . . |   ∃e':E.
| . . . . |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
| . . . . 28. LL :
| . . . . |   n:ℕ||sends(l;e)||→{e':E
| . . . . |                       | isrcv(e')
| . . . . |                         & lnk(e') = l
| . . . . |                           & sender(e') = e
| . . . . |                           & index(e') = n }
| . . . . 29. (λn.1of(L(n))) = LL
```

```
| . . . . 30. (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . . . 31. X :
| . . . . |   {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . . . 32. map(LL;upto(||sends(l;e)||)) = X
| . . . . 33. e' : E
| . . . . 34. isrcv(e')
| . . . . 35. lnk(e') = l
| . . . . 36. sender(e') = e
| . . . . ⊢  valtype(e') ⊆r ma-valtype(da; rcv(l; tag(e')))
| . . . . | by Subst' (rcv(l; tag(e')) = kind(e')) 0
| . . . . |
| . . . . |\
| . . . . | 1. i : Id
| . . . . | 2. k : Knd
| . . . . | 3. l : IdLnk
| . . . . | 4. ds : x:Id fp-> Type
| . . . . | 5. da : a:Knd fp-> Type
| . . . . | 6. f :
| . . . . |    (tg:Id
| . . . . |     ×State(ds)→ma-valtype(da; k)→
| . . . . |     (da(rcv(l; tg))?Void List)) List
| . . . . | 7. source(l) = i
| . . . . | 8. D' : Dsys
| . . . . | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . . . | 10. w : World
| . . . . | 11. p : FairFifo
| . . . . | 12. PossibleWorld(D';w)
| . . . . | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . . | 14. ∀e:E.
| . . . . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . | 15. ∀e:E.
| . . . . |     isrcv(e)
| . . . . |     ⇒
| . . . . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . . . |     ((λl,tg. da(rcv(l; tg))?Top))
| . . . . | 17. ∀e:E.
| . . . . |     loc(e) = i
| . . . . |     ⇒
| . . . . |     kind(e) = k
| . . . . |     ⇒
| . . . . |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . . | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . . | 19. ∀e:E.
| . . . . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . | 20. ∀e:E.
| . . . . |     isrcv(e)
| . . . . |     ⇒
| . . . . |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . | 21. e : E
| . . . . | 22. loc(e) = i
| . . . . | 23. kind(e) = k
| . . . . | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . . | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . . | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
```

```
| . . . . | 27. L :
| . . . . |     ∀n:ℕ||sends(l;e)||.
| . . . . |     ∃e':E.
| . . . . |     isrcv(e')
| . . . . |     & lnk(e') = l & sender(e') = e & index(e') = n
| . . . . | 28. LL :
| . . . . |     n:ℕ||sends(l;e)||→{e':E
| . . . . |                         | isrcv(e')
| . . . . |                           & lnk(e') = l
| . . . . |                             & sender(e') = e
| . . . . |                             & index(e') = n }
| . . . . | 29. (λn.1of(L(n))) = LL
| . . . . | 30. (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . . . | 31. X :
| . . . . |     {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . . . | 32. map(LL;upto(||sends(l;e)||)) = X
| . . . . | 33. e' : E
| . . . . | 34. isrcv(e')
| . . . . | 35. lnk(e') = l
| . . . . | 36. sender(e') = e
| . . . . | ⊢  rcv(l; tag(e')) = kind(e')
| . . . . |   by MoveToConcl -2 THEN MoveToConcl -2
| . . . . |      THEN
| . . . . |      Unfolds [`es-isrcv`;`es-lnk`;`es-tag`] 0
| . . . . |      THEN
| . . . . |      GenConclAtAddr [2;2;3]
| . . . . |      THEN
| . . . . |      Analyze -2
| . . . . |      THEN
| . . . . |      Analyze -2
| . . . . |      THEN
| . . . . |      Repeat
| . . . . |      (Unfolds [`isrcv`;`lnk`;`tagof`] 0 THEN Reduce 0)
| . . . . |      THEN
| . . . . |      Fold `rcv` 0
| . . . . |      THEN
| . . . . |      Analyze
| . . . . |
| . . . . .\
| . . . . . 1. i : Id
| . . . . . 2. k : Knd
| . . . . . 3. l : IdLnk
| . . . . . 4. ds : x:Id fp-> Type
| . . . . . 5. da : a:Knd fp-> Type
| . . . . . 6. f :
| . . . . .    (tg:Id
| . . . . .     ×State(ds)→ma-valtype(da; k)→
| . . . . .     (da(rcv(l; tg))?Void List)) List
| . . . . . 7. source(l) = i
| . . . . . 8. D' : Dsys
| . . . . . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
| . . . . . 10. w : World
| . . . . . 11. p : FairFifo
| . . . . . 12. PossibleWorld(D';w)
| . . . . . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
```

```
| . . . . . 14. ∀e:E.
| . . . . .     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . . 15. ∀e:E.
| . . . . .     isrcv(e)
| . . . . .     ⇒
| . . . . .     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
| . . . . .     ((λl,tg. da(rcv(l; tg))?Top))
| . . . . . 17. ∀e:E.
| . . . . .     loc(e) = i
| . . . . .     ⇒
| . . . . .     kind(e) = k
| . . . . .     ⇒
| . . . . .     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . . . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . . . . 19. ∀e:E.
| . . . . .     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . . 20. ∀e:E.
| . . . . .     isrcv(e)
| . . . . .     ⇒
| . . . . .     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . . . . 21. e : E
| . . . . . 22. loc(e) = i
| . . . . . 23. kind(e) = k
| . . . . . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . . . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
| . . . . . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
| . . . . . 27. L :
| . . . . .     ∀n:ℕ||sends(l;e)||.
| . . . . .     ∃e':E.
| . . . . .     isrcv(e')
| . . . . .     & lnk(e') = l & sender(e') = e & index(e') = n
| . . . . . 28. LL :
| . . . . .     n:ℕ||sends(l;e)||→{e':E
| . . . . .                          | isrcv(e')
| . . . . .                            & lnk(e') = l
| . . . . .                              & sender(e') = e
| . . . . .                              & index(e') = n }
| . . . . . 29. (λn.1of(L(n))) = LL
| . . . . . 30. (tg:Id×ma-valtype(da; rcv(l; tg))) List
| . . . . . 31. X :
| . . . . .     {e':E| isrcv(e') & lnk(e') = l & sender(e') = e } List
| . . . . . 32. map(LL;upto(||sends(l;e)||)) = X
| . . . . . 33. e' : E
| . . . . . 34. isrcv(e')
| . . . . . 35. lnk(e') = l
| . . . . . 36. sender(e') = e

| . . . . . ⊢  valtype(e') ⊆r ma-valtype(da; kind(e'))
| . . . . .   by BackThruSomeHyp
|
.\
. .....wf .....
. 1. i : Id
. 2. k : Knd
. 3. l : IdLnk
```

```
. 4. ds : x:Id fp-> Type
. 5. da : a:Knd fp-> Type
. 6. f :
. |  (tg:Id
. |   ×State(ds)→ma-valtype(da; k)→(da(rcv(l; tg))?Void List)) List
. 7. source(l) = i
. 8. D' : Dsys
. 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
. 10. w : World
. 11. p : FairFifo
. 12. PossibleWorld(D';w)
. 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. 15. ∀e:E.
. |   isrcv(e)
. |   ⇒
. |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
. |   ((λl,tg. da(rcv(l; tg))?Top))
. 17. ∀e:E.
. |   loc(e) = i
. |   ⇒
. |   kind(e) = k
. |   ⇒
. |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. 20. ∀e:E.
. |   isrcv(e)
. |   ⇒
. |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. 21. e : E
. 22. loc(e) = i
. 23. kind(e) = k
. 24. valtype(e) ⊆r ma-valtype(da; kind(e))
. 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. 26. valtype(e) ⊆r ma-valtype(da; kind(e))
. 27. ∀n:ℕ||sends(l;e)||.
. |   ∃e':E. isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
. 28. L1 : {e':E| isrcv(e') & lnk(e') = l } List
. ⊢  ((∀e':E. (e' ∈ L1) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e)
. |   & (∀e1,e2:E. e1 before e2 ∈ L1 ⇒ (e1 <loc e2))
. |   & map(λe'.<tag(e'),val(e')>;L1)
. |     =
. |     tagged-list-messages(λz.(z when e);val(e);f))
. |  ∈ Prop  by Auto
. |
. |\
. | 1. i : Id
. | 2. k : Knd
. | 3. l : IdLnk
. | 4. ds : x:Id fp-> Type
. | 5. da : a:Knd fp-> Type
. | 6. f :
. |    (tg:Id
```

```
. |     ×State(ds)→ma-valtype(da; k)→
. |     (da(rcv(l; tg))?Void List)) List
. | 7. source(l) = i
. | 8. D' : Dsys
. | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
. | 10. w : World
. | 11. p : FairFifo
. | 12. PossibleWorld(D';w)
. | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 15. ∀e:E.
. |     isrcv(e)
. |     ⇒
. |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
. |     ((λl,tg. da(rcv(l; tg))?Top))
. | 17. ∀e:E.
. |     loc(e) = i
. |     ⇒
. |     kind(e) = k
. |     ⇒
. |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 20. ∀e:E.
. |     isrcv(e)
. |     ⇒
. |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 21. e : E
. | 22. loc(e) = i
. | 23. kind(e) = k
. | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
. | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
. | 27. ∀n:ℕ||sends(l;e)||.
. |     ∃e':E.
. |     isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
. | 28. {e':E| isrcv(e') & lnk(e') = l } List
. | 29. e' : {e':E| isrcv(e') & lnk(e') = l }
. | ⊢  isrcv(e')  by Analyze -1 THEN Unhide THEN ExRepD
. |
. |\
. | .....subterm .....
. | 1. i : Id
. | 2. k : Knd
. | 3. l : IdLnk
. | 4. ds : x:Id fp-> Type
. | 5. da : a:Knd fp-> Type
. | 6. f :
. | |  (tg:Id
. | |   ×State(ds)→ma-valtype(da; k)→
. | |   (da(rcv(l; tg))?Void List)) List
. | 7. source(l) = i
. | 8. D' : Dsys
. | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
```

```
. | 10. w : World
. | 11. p : FairFifo
. | 12. PossibleWorld(D';w)
. | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 15. ∀e:E.
. | |   isrcv(e)
. | |   ⇒
. | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
. | |   ((λl,tg. da(rcv(l; tg))?Top))
. | 17. ∀e:E.
. | |   loc(e) = i
. | |   ⇒
. | |   kind(e) = k
. | |   ⇒
. | |   sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 20. ∀e:E.
. | |   isrcv(e)
. | |   ⇒
. | |   lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 21. e : E
. | 22. loc(e) = i
. | 23. kind(e) = k
. | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
. | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
. | 27. ∀n:ℕ||sends(l;e)||.
. | |   ∃e':E.
. | |   isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
. | 28. {e':E| isrcv(e') & lnk(e') = l } List
. | 29. e' : {e':E| isrcv(e') & lnk(e') = l }
. | ⊢  val(e') ∈ ma-valtype(da; rcv(l; tag(e')))
. | | by ExRepD THEN DoSubsume
. | |    THEN
. | |    Subst' (rcv(l; tag(e')) = kind(e')) 0
. | |    THEN
. | |    Analyze -1
. | |    THEN
. | |    Unhide
. | |    THEN
. | |    ExRepD
. | |    THEN
. | |    Try BackThruSomeHyp
. | |
. | 1. i : Id
. | 2. k : Knd
. | 3. l : IdLnk
. | 4. ds : x:Id fp-> Type
. | 5. da : a:Knd fp-> Type
. | 6. f :
. |    (tg:Id
. |     ×State(ds)→ma-valtype(da; k)→
```

```
. |     (da(rcv(l; tg))?Void List)) List
. | 7. source(l) = i
. | 8. D' : Dsys
. | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
. | 10. w : World
. | 11. p : FairFifo
. | 12. PossibleWorld(D';w)
. | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 15. ∀e:E.
. |     isrcv(e)
. |     ⇒
. |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
. |     ((λl,tg. da(rcv(l; tg))?Top))
. | 17. ∀e:E.
. |     loc(e) = i
. |     ⇒
. |     kind(e) = k
. |     ⇒
. |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 20. ∀e:E.
. |     isrcv(e)
. |     ⇒
. |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 21. e : E
. | 22. loc(e) = i
. | 23. kind(e) = k
. | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
. | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
. | 27. ∀n:ℕ||sends(l;e)||.
. |     ∃e':E.
. |     isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
. | 28. {e':E| isrcv(e') & lnk(e') = l } List
. | 29. e' : E
. | 30. isrcv(e')
. | 31. lnk(e') = l
. | ⊢  rcv(l; tag(e')) = kind(e')
. |   by MoveToConcl -1 THEN MoveToConcl -1
. |      THEN
. |      Unfolds [`es-isrcv`;`es-lnk`;`es-tag`] 0
. |      THEN
. |      GenConclAtAddr [1;1;1]
. |      THEN
. |      Analyze -2
. |      THEN
. |      Analyze -2
. |      THEN
. |      Repeat (Unfolds [`isrcv`;`lnk`;`tagof`] 0 THEN Reduce 0)
. |      THEN
. |      Analyze 0
. |      THEN
```

```
. |      Try Trivial
. |      THEN
. |      Fold ‘rcv‘ 0
. |      THEN
. |      Analyze
. |
. |\
. | .....subterm .....
. | 1. i : Id
. | 2. k : Knd
. | 3. l : IdLnk
. | 4. ds : x:Id fp-> Type
. | 5. da : a:Knd fp-> Type
. | 6. f :
. |    (tg:Id
. |     ×State(ds)→ma-valtype(da; k)→
. |     (da(rcv(l; tg))?Void List)) List
. | 7. source(l) = i
. | 8. D’ : Dsys
. | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D’
. | 10. w : World
. | 11. p : FairFifo
. | 12. PossibleWorld(D’;w)
. | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 15. ∀e:E.
. |     isrcv(e)
. |     ⇒
. |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
. |     ((λl,tg. da(rcv(l; tg))?Top))
. | 17. ∀e:E.
. |     loc(e) = i
. |     ⇒
. |     kind(e) = k
. |     ⇒
. |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 20. ∀e:E.
. |     isrcv(e)
. |     ⇒
. |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 21. e : E
. | 22. loc(e) = i
. | 23. kind(e) = k
. | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
. | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
. | 27. ∀n:ℕ||sends(l;e)||.
. |     ∃e’:E.
. |     isrcv(e’) & lnk(e’) = l & sender(e’) = e & index(e’) = n
. | 28. {e’:E| isrcv(e’) & lnk(e’) = l } List
. | ⊢  (λz.(z when e)) ∈ State(ds)
. |   by Unfold ‘ma-state‘ 0 THEN DoSubsume THEN Subst (loc(e) = i) 0
```

```
. |      THEN
. |      BackThruSomeHyp
. |
. |\
. | .....subterm .....
. | 1. i : Id
. | 2. k : Knd
. | 3. l : IdLnk
. | 4. ds : x:Id fp-> Type
. | 5. da : a:Knd fp-> Type
. | 6. f :
. |    (tg:Id
. |     ×State(ds)→ma-valtype(da; k)→
. |     (da(rcv(l; tg))?Void List)) List
. | 7. source(l) = i
. | 8. D' : Dsys
. | 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
. | 10. w : World
. | 11. p : FairFifo
. | 12. PossibleWorld(D';w)
. | 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. | 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 15. ∀e:E.
. |     isrcv(e)
. |     ⇒
. |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
. |     ((λl,tg. da(rcv(l; tg))?Top))
. | 17. ∀e:E.
. |     loc(e) = i
. |     ⇒
. |     kind(e) = k
. |     ⇒
. |     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. | 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. | 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 20. ∀e:E.
. |     isrcv(e)
. |     ⇒
. |     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. | 21. e : E
. | 22. loc(e) = i
. | 23. kind(e) = k
. | 24. valtype(e) ⊆r ma-valtype(da; kind(e))
. | 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. | 26. valtype(e) ⊆r ma-valtype(da; kind(e))
. | 27. ∀n:ℕ||sends(l;e)||.
. |     ∃e':E.
. |     isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
. | 28. {e':E| isrcv(e') & lnk(e') = l } List
. | ⊢  val(e) ∈ ma-valtype(da; k)
. |   by DoSubsume THEN ExRepD
. |      THEN
. |      AllHyps (λh.InstHyp [e] h THENA Complete Auto)
. |      THEN
```

```
. |      NthHypEq -1
. |      THEN
. |      Repeat Analyze
. |
. .\
. . .....unproved Inclusion subgoal .....
. . 1. i : Id
. . 2. k : Knd
. . 3. l : IdLnk
. . 4. ds : x:Id fp-> Type
. . 5. da : a:Knd fp-> Type
. . 6. f :
. .    (tg:Id
. .     ×State(ds)→ma-valtype(da; k)→
. .     (da(rcv(l; tg))?Void List)) List
. . 7. source(l) = i
. . 8. D' : Dsys
. . 9. d-single-sends(i; ds; da; k; l; f) ⊆ D'
. . 10. w : World
. . 11. p : FairFifo
. . 12. PossibleWorld(D';w)
. . 13. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . 14. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . 15. ∀e:E.
. .     isrcv(e)
. .     ⇒
. .     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . 16. {m:Msg| source(mlnk(m)) = i } ⊆r Msg
. .     ((λl,tg. da(rcv(l; tg))?Top))
. . 17. ∀e:E.
. .     loc(e) = i
. .     ⇒
. .     kind(e) = k
. .     ⇒
. .     sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. . 18. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . 19. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . 20. ∀e:E.
. .     isrcv(e)
. .     ⇒
. .     lnk(e) = l ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . 21. e : E
. . 22. loc(e) = i
. . 23. kind(e) = k
. . 24. valtype(e) ⊆r ma-valtype(da; kind(e))
. . 25. sends(l;e) = tagged-messages(l;λz.(z when e);val(e);f)
. . 26. valtype(e) ⊆r ma-valtype(da; kind(e))
. . 27. ∀n:ℕ||sends(l;e)||.
. .     ∃e':E.
. .     isrcv(e') & lnk(e') = l & sender(e') = e & index(e') = n
. . 28. {e':E| isrcv(e') & lnk(e') = l } List
. . 29. (tg:Id
. .      ×State(ds)→ma-valtype(da; k)→
. .      (da(rcv(l; tg))?Void List)) List
. . 30. tg:Id×State(ds)→ma-valtype(da; k)→(da(rcv(l; tg))?Void List)
```

```
. . 31. tg : Id
. . 32. State(ds)
. . 33. ma-valtype(da; k)
. . 34. da(rcv(l; tg))?Void List
. . 35. u1 : da(rcv(l; tg))?Void
. . ⊢  u1 ∈ ma-valtype(da; rcv(l; tg))
. .    by DoSubsume THEN Unfold ‘ma-valtype‘ 0
```

END