```
s-sends-rule1  THM

⊢  ∀x,tg:Id, k:Knd, l:IdLnk, T,A,B:Type, f:(A→B→(T List)).
|  (rcv(l; tg) = k ⇒ T = B)
|  ⇒
|  @source(l): ma-single-sends1(A; B; T; x; k; l; tg; f) ∈ Dsys
|  & (∀D:Dsys.
|     @source(l): ma-single-sends1(A; B; T; x; k; l; tg; f) ⊆ D
|     ⇒
|     D
|     realizes es.(vartype(source(l);x) ⊆r A)
|                 & (∀e:E.
|                    loc(e) = source(l)
|                    ⇒
|                    kind(e) = k ⇒ (valtype(e) ⊆r B))
|                 & (∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T))
|                 & (∀e:E.
|                    loc(e) = source(l)
|                    ⇒
|                    kind(e) = k
|                    ⇒
|                    (∃L:{e':E| kind(e') = rcv(l; tg) } List.
|                     (∀e':E.
|                      (e' ∈ L)
|                      ⟺
|                      kind(e') = rcv(l; tg) & sender(e') = e)
|                     & (∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2))
|                     & map(λe'.val(e');L) = f((x when e),val(e)))))
| by UnivCD THEN Unfold `ma-single-sends1` 0
|    THEN
|    Inst Thm: s_DASH_sends_DASH_rule
|    [source(l)
|    ;k
|    ;l
|    ;x : A
|    ;k : B ⊕ rcv(l; tg) : T
|    ;[<tg,λs,v. f(s(x),v)>]]
|    THENA
|    Try (Complete Auto)
|
|\
| .....wf .....
| 1. x : Id
| 2. tg : Id
| 3. k : Knd
| 4. l : IdLnk
| 5. T : Type
| 6. A : Type
| 7. B : Type
| 8. f : A→B→(T List)
| 9. rcv(l; tg) = k ⇒ T = B
| ⊢  [<tg,λs,v. f(s(x),v)>]
| |  ∈ (tg@0:Id
| |     ×State(x : A)→ma-valtype(k : B ⊕ rcv(l; tg) : T; k)→
| |     (k : B ⊕ rcv(l; tg) : T(rcv(l; tg@0))?Void List)) List
```

```
| | by AssertBY (x : A ∈ x:Id fp-> Type) Auto
| |    THEN
| |    AssertBY (k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type) Auto
| |    THEN
| |    At Type Auto
| |    THEN
| |    Try (Fold `ma-state` 0)
| |    THEN
| |    DoSubsume
| |
| |\
| | .....subtype_rel .....
| | 1. x : Id
| | 2. tg : Id
| | 3. k : Knd
| | 4. l : IdLnk
| | 5. T : Type
| | 6. A : Type
| | 7. B : Type
| | 8. A→B→(T List)
| | 9. rcv(l; tg) = k ⇒ T = B
| | 10. x : A ∈ x:Id fp-> Type
| | 11. k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type
| | 12. State(x : A)
| | 13. ma-valtype(k : B ⊕ rcv(l; tg) : T; k)
| | ⊢  ma-valtype(k : B ⊕ rcv(l; tg) : T; k) ⊆r B
| |   by Unfold `ma-valtype` 0
| |      THEN
| |      RWO
| |      Thm* ∀eq:EqDecider(A), f,g:a:A fp-> Top, x:A, z:Top.
| |           f ⊕ g(x)?z ∼ if x ∈ dom(f)→ f(x)?z else g(x)?z fi
| |      {fpf_DASH_join_DASH_cap_DASH_sq}
| |      0
| |      THEN
| |      Reduce 0
| |      THEN
| |      SplitOnConclITE
| |      THEN
| |      Analyze -1
| |      THEN
| |      RWO
| |      Thm* ∀eq:EqDecider(A), x,y:A, v:Top. x ∈ dom(y : v) ⟺ x = y
| |      {fpf_DASH_single_DASH_dom}
| |      0
| |
| .\
| . .....subtype_rel .....
| . 1. x : Id
| . 2. tg : Id
| . 3. k : Knd
| . 4. l : IdLnk
| . 5. T : Type
| . 6. A : Type
| . 7. B : Type
| . 8. f : A→B→(T List)
```

```
| . 9. rcv(l; tg) = k ⇒ T = B
| . 10. x : A ∈ x:Id fp-> Type
| . 11. k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type
| . 12. s : State(x : A)
| . 13. v : ma-valtype(k : B ⊕ rcv(l; tg) : T; k)
| . 14. f(s(x),v) = f(s(x),v)
| . 15. T List
| . 16. T List
| . 17. T
| . ⊢  T ⊆r k : B ⊕ rcv(l; tg) : T(rcv(l; tg))?Void
| . | by RWO
| . |    Thm* ∀eq:EqDecider(A), f,g:a:A fp-> Top, x:A, z:Top.
| . |         f ⊕ g(x)?z ∼ if x ∈ dom(f)→ f(x)?z else g(x)?z fi
| . |    {fpf_DASH_join_DASH_cap_DASH_sq}
| . |    0
| . |    THEN
| . |    SplitOnConclITE
| . |    THEN
| . |    Reduce 0
| . |    THEN
| . |    RWO
| . |    Thm* ∀eq:EqDecider(A), x,y:A, v:Top. x ∈ dom(y : v) ⟺ x = y
| . |    {fpf_DASH_single_DASH_dom}
| . |    -1
| . |    THEN
| . |    AssertBY (k : B ∈ k:Knd fp-> Type) Auto
| . |    THEN
| . |    HypSubst -2 0
| . |    THEN
| . |    Reduce 0
| . |
| . 1. x : Id
| . 2. tg : Id
| . 3. k : Knd
| . 4. l : IdLnk
| . 5. T : Type
| . 6. A : Type
| . 7. B : Type
| . 8. f : A→B→(T List)
| . 9. rcv(l; tg) = k ⇒ T = B
| . 10. x : A ∈ x:Id fp-> Type
| . 11. k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type
| . 12. s : State(x : A)
| . 13. v : ma-valtype(k : B ⊕ rcv(l; tg) : T; k)
| . 14. f(s(x),v) = f(s(x),v)
| . 15. T List
| . 16. T List
| . 17. T
| . 18. rcv(l; tg) = k
| . 19. k : B ∈ k:Knd fp-> Type
| . ⊢  T ⊆r B  by SubtypeRelEq THEN BackThruSomeHyp
|
.\
. 1. x : Id
. 2. tg : Id
```

```
. 3. k : Knd
. 4. l : IdLnk
. 5. T : Type
. 6. A : Type
. 7. B : Type
. 8. f : A→B→(T List)
. 9. rcv(l; tg) = k ⇒ T = B
. 10. @source(l): ma-single-sends(x : A;
. |                               k : B ⊕ rcv(l; tg) : T;
. |                               k;
. |                               l;
. |                               [<tg,λs,v. f(s(x),v)>])
. |    ∈ Dsys
. |    & (∀D:Dsys.
. |       @source(l): ma-single-sends(x : A;
. |                                    k : B ⊕ rcv(l; tg) : T;
. |                                    k;
. |                                    l;
. |                                    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. |       ⇒
. |       D
. |       realizes es.(∀x@0:Id.
. |                     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. |                   & (∀e:E.
. |                      loc(e) = source(l)
. |                      ⇒
. |                      (valtype(e) ⊆r ma-valtype
. |                       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. |                   & (∀e:E.
. |                      isrcv(e)
. |                      ⇒
. |                      lnk(e) = l
. |                      ⇒
. |                      (valtype(e) ⊆r ma-valtype
. |                       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. |                   & (∀e:E.
. |                      loc(e) = source(l)
. |                      ⇒
. |                      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);[<tg
. |                                            ,λs,v. f(s(x),v)>])))))
. ⊢  @source(l): ma-single-sends(x : A;
. |                              k : B ⊕ rcv(l; tg) : T;
. |                              k;
```

```
. |                                l;
. |                                [<tg,λs,v. f(s(x),v)>])
. |  ∈ Dsys
. |  & (∀D:Dsys.
. |     @source(l): ma-single-sends(x : A;
. |                                  k : B ⊕ rcv(l; tg) : T;
. |                                  k;
. |                                  l;
. |                                  [<tg,λs,v. f(s(x),v)>]) ⊆ D
. |     ⇒
. |     D
. |     realizes es.(vartype(source(l);x) ⊆r A)
. |                 & (∀e:E.
. |                    loc(e) = source(l)
. |                    ⇒
. |                    kind(e) = k ⇒ (valtype(e) ⊆r B))
. |                 & (∀e:E.
. |                    kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T))
. |                 & (∀e:E.
. |                    loc(e) = source(l)
. |                    ⇒
. |                    kind(e) = k
. |                    ⇒
. |                    (∃L:{e':E| kind(e') = rcv(l; tg) } List.
. |                     (∀e':E.
. |                      (e' ∈ L)
. |                      ⟺
. |                      kind(e') = rcv(l; tg) & sender(e') = e)
. |                     & (∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2))
. |                     & map(λe'.val(e');L) = f((x when e),val(e)))))
. | by ExRepD THEN Analyze 0 THEN Try Trivial THEN Parallel0p -2
. |    THEN
. |    RepeatFor 7 (Parallel0p -1)
. |    THEN
. |    ExRepD
. |    THEN
. |    BetterSplitAndConcl
. |
. |\
. | 1. x : Id
. | 2. tg : Id
. | 3. k : Knd
. | 4. l : IdLnk
. | 5. T : Type
. | 6. A : Type
. | 7. B : Type
. | 8. f : A→B→(T List)
. | 9. rcv(l; tg) = k ⇒ T = B
. | 10. @source(l): ma-single-sends(x : A;
. |                                 k : B ⊕ rcv(l; tg) : T;
. |                                 k;
. |                                 l;
. |                                 [<tg,λs,v. f(s(x),v)>])
. |     ∈ Dsys
. | 11. ∀D:Dsys.
```

```
. |     @source(l): ma-single-sends(x : A;
. |                                  k : B ⊕ rcv(l; tg) : T;
. |                                  k;
. |                                  l;
. |                                  [<tg,λs,v. f(s(x),v)>]) ⊆ D
. |     ⇒
. |     D
. |     realizes es.(∀x@0:Id.
. |                   vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. |                  & (∀e:E.
. |                     loc(e) = source(l)
. |                     ⇒
. |                     (valtype(e) ⊆r ma-valtype
. |                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. |                  & (∀e:E.
. |                     isrcv(e)
. |                     ⇒
. |                     lnk(e) = l
. |                     ⇒
. |                     (valtype(e) ⊆r ma-valtype
. |                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. |                  & (∀e:E.
. |                     loc(e) = source(l)
. |                     ⇒
. |                     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);[<tg,λs,v. f(s(x),v)>])))
. | 12. @source(l): ma-single-sends(x : A;
. |                                  k : B ⊕ rcv(l; tg) : T;
. |                                  k;
. |                                  l;
. |                                  [<tg,λs,v. f(s(x),v)>])
. |     ∈ Dsys
. | 13. D : Dsys
. | 14. @source(l): ma-single-sends(x : A;
. |                                  k : B ⊕ rcv(l; tg) : T;
. |                                  k;
. |                                  l;
. |                                  [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | 15. ∀D':Dsys.
. |     D ⊆ D'
. |     ⇒
. |     (∀w:World, p:FairFifo.
. |      PossibleWorld(D';w)
. |      ⇒
. |      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
```

```
. |      & (∀e:E.
. |         loc(e) = source(l)
. |         ⇒
. |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. |                                    )))
. |      & (∀e:E.
. |         isrcv(e)
. |         ⇒
. |         lnk(e) = l
. |         ⇒
. |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. |                                    )))
. |      & (∀e:E.
. |         loc(e) = source(l)
. |         ⇒
. |         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);[<tg,λs,v. f(s(x),v)>])))))
. | 16. D' : Dsys
. | 17. D ⊆ D'
. | 18. ∀w:World, p:FairFifo.
. |     PossibleWorld(D';w)
. |     ⇒
. |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. |     & (∀e:E.
. |        loc(e) = source(l)
. |        ⇒
. |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. |                                   )))
. |     & (∀e:E.
. |        isrcv(e)
. |        ⇒
. |        lnk(e) = l
. |        ⇒
. |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. |                                   )))
. |     & (∀e:E.
. |        loc(e) = source(l)
. |        ⇒
. |        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);[<tg,λs,v. f(s(x),v)>])))
. | 19. w : World
. | 20. ∀p:FairFifo.
. |     PossibleWorld(D';w)
. |     ⇒
. |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. |     & (∀e:E.
. |        loc(e) = source(l)
. |        ⇒
. |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. |                                  )))
. |     & (∀e:E.
. |        isrcv(e)
. |        ⇒
. |        lnk(e) = l
. |        ⇒
. |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. |                                  )))
. |     & (∀e:E.
. |        loc(e) = source(l)
. |        ⇒
. |        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);[<tg,λs,v. f(s(x),v)>])))
. | 21. p : FairFifo
. | 22. PossibleWorld(D';w)
. | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | 24. ∀e:E.
. |     loc(e) = source(l)
. |     ⇒
. |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | 25. ∀e:E.
. |     isrcv(e)
. |     ⇒
. |     lnk(e) = l
. |     ⇒
. |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | 26. ∀e:E.
. |     loc(e) = source(l)
. |     ⇒
. |     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);[<tg
```

```
. |                                                    ,λs,v. f
. |                                                           (s(x)
. |                                                           ,v)>]))
. | ⊢  vartype(source(l);x) ⊆r A  by InstHyp [x] -4 THEN Reduce -1
. |
. |\
. | 1. x : Id
. | 2. tg : Id
. | 3. k : Knd
. | 4. l : IdLnk
. | 5. T : Type
. | 6. A : Type
. | 7. B : Type
. | 8. f : A→B→(T List)
. | 9. rcv(l; tg) = k ⇒ T = B
. | 10. @source(l): ma-single-sends(x : A;
. | |                               k : B ⊕ rcv(l; tg) : T;
. | |                               k;
. | |                               l;
. | |                               [<tg,λs,v. f(s(x),v)>])
. | |   ∈ Dsys
. | 11. ∀D:Dsys.
. | |   @source(l): ma-single-sends(x : A;
. | |                               k : B ⊕ rcv(l; tg) : T;
. | |                               k;
. | |                               l;
. | |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | |   ⇒
. | |   D
. | |   realizes es.(∀x@0:Id.
. | |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |                & (∀e:E.
. | |                   loc(e) = source(l)
. | |                   ⇒
. | |                   (valtype(e) ⊆r ma-valtype
. | |                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | |                & (∀e:E.
. | |                   isrcv(e)
. | |                   ⇒
. | |                   lnk(e) = l
. | |                   ⇒
. | |                   (valtype(e) ⊆r ma-valtype
. | |                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | |                & (∀e:E.
. | |                   loc(e) = source(l)
. | |                   ⇒
. | |                   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);[<tg,λs,v. f(s(x),v)>])))
. | 12. @source(l): ma-single-sends(x : A;
. | |                               k : B ⊕ rcv(l; tg) : T;
. | |                               k;
. | |                               l;
. | |                               [<tg,λs,v. f(s(x),v)>])
. | |   ∈ Dsys
. | 13. D : Dsys
. | 14. @source(l): ma-single-sends(x : A;
. | |                               k : B ⊕ rcv(l; tg) : T;
. | |                               k;
. | |                               l;
. | |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | 15. ∀D':Dsys.
. | |   D ⊆ D'
. | |   ⇒
. | |   (∀w:World, p:FairFifo.
. | |    PossibleWorld(D';w)
. | |    ⇒
. | |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |    & (∀e:E.
. | |       loc(e) = source(l)
. | |       ⇒
. | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                  )))
. | |    & (∀e:E.
. | |       isrcv(e)
. | |       ⇒
. | |       lnk(e) = l
. | |       ⇒
. | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                  )))
. | |    & (∀e:E.
. | |       loc(e) = source(l)
. | |       ⇒
. | |       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);[<tg,λs,v. f(s(x),v)>])))))
. | 16. D' : Dsys
. | 17. D ⊆ D'
. | 18. ∀w:World, p:FairFifo.
. | |   PossibleWorld(D';w)
. | |   ⇒
. | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |   & (∀e:E.
. | |      loc(e) = source(l)
```

```
. | |      ⇒
. | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                 )))
. | |   & (∀e:E.
. | |      isrcv(e)
. | |      ⇒
. | |      lnk(e) = l
. | |      ⇒
. | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                 )))
. | |   & (∀e:E.
. | |      loc(e) = source(l)
. | |      ⇒
. | |      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);[<tg,λs,v. f(s(x),v)>])))
. | 19. w : World
. | 20. ∀p:FairFifo.
. | |   PossibleWorld(D';w)
. | |   ⇒
. | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |   & (∀e:E.
. | |      loc(e) = source(l)
. | |      ⇒
. | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                 )))
. | |   & (∀e:E.
. | |      isrcv(e)
. | |      ⇒
. | |      lnk(e) = l
. | |      ⇒
. | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                 )))
. | |   & (∀e:E.
. | |      loc(e) = source(l)
. | |      ⇒
. | |      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);[<tg,λs,v. f(s(x),v)>])))
. | 21. p : FairFifo
. | 22. PossibleWorld(D';w)
```

```
. | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | 24. ∀e:E.
. | |   loc(e) = source(l)
. | |   ⇒
. | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | 25. ∀e:E.
. | |   isrcv(e)
. | |   ⇒
. | |   lnk(e) = l
. | |   ⇒
. | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | 26. ∀e:E.
. | |   loc(e) = source(l)
. | |   ⇒
. | |   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);[<tg
. | |                                                 ,λs,v. f
. | |                                                        (s(x)
. | |                                                        ,v)>]))
. | 27. vartype(source(l);x) ⊆r A
. | ⊢  ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | | by Auto THEN InstHyp [e] -7 THEN NthHypEq -1 THEN Analyze
. | |    THEN
. | |    HypSubst -2 0
. | |    THEN
. | |    Unfold `ma-valtype` 0
. | |    THEN
. | |    RWO
. | |    Thm* ∀eq:EqDecider(A), f,g:a:A fp-> Top, x:A, z:Top.
. | |         f ⊕ g(x)?z ∼ if x ∈ dom(f)→ f(x)?z else g(x)?z fi
. | |    {fpf_DASH_join_DASH_cap_DASH_sq}
. | |    0
. | |    THEN
. | |    Reduce 0
. | |    THEN
. | |    SplitOnConclITE
. | |
. | .....falsecase .....
. | 1. x : Id
. | 2. tg : Id
. | 3. k : Knd
. | 4. l : IdLnk
. | 5. T : Type
. | 6. A : Type
. | 7. B : Type
. | 8. f : A→B→(T List)
. | 9. rcv(l; tg) = k ⇒ T = B
. | 10. @source(l): ma-single-sends(x : A;
. |                                 k : B ⊕ rcv(l; tg) : T;
```

```
. |                                k;
. |                                l;
. |                                [<tg,λs,v. f(s(x),v)>])
. |     ∈ Dsys
. | 11. ∀D:Dsys.
. |     @source(l): ma-single-sends(x : A;
. |                                k : B ⊕ rcv(l; tg) : T;
. |                                k;
. |                                l;
. |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. |     ⇒
. |     D
. |     realizes es.(∀x@0:Id.
. |                   vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. |                  & (∀e:E.
. |                     loc(e) = source(l)
. |                     ⇒
. |                     (valtype(e) ⊆r ma-valtype
. |                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. |                  & (∀e:E.
. |                     isrcv(e)
. |                     ⇒
. |                     lnk(e) = l
. |                     ⇒
. |                     (valtype(e) ⊆r ma-valtype
. |                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. |                  & (∀e:E.
. |                     loc(e) = source(l)
. |                     ⇒
. |                     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);[<tg,λs,v. f(s(x),v)>])))
. | 12. @source(l): ma-single-sends(x : A;
. |                                k : B ⊕ rcv(l; tg) : T;
. |                                k;
. |                                l;
. |                                [<tg,λs,v. f(s(x),v)>])
. |     ∈ Dsys
. | 13. D : Dsys
. | 14. @source(l): ma-single-sends(x : A;
. |                                k : B ⊕ rcv(l; tg) : T;
. |                                k;
. |                                l;
. |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | 15. ∀D':Dsys.
. |     D ⊆ D'
```

```
. |     ⇒
. |     (∀w:World, p:FairFifo.
. |      PossibleWorld(D';w)
. |      ⇒
. |      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. |      & (∀e:E.
. |         loc(e) = source(l)
. |         ⇒
. |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. |                                   )))
. |      & (∀e:E.
. |         isrcv(e)
. |         ⇒
. |         lnk(e) = l
. |         ⇒
. |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. |                                   )))
. |      & (∀e:E.
. |         loc(e) = source(l)
. |         ⇒
. |         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);[<tg,λs,v. f(s(x),v)>])))))
. | 16. D' : Dsys
. | 17. D ⊆ D'
. | 18. ∀w:World, p:FairFifo.
. |     PossibleWorld(D';w)
. |     ⇒
. |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. |     & (∀e:E.
. |        loc(e) = source(l)
. |        ⇒
. |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. |                                   )))
. |     & (∀e:E.
. |        isrcv(e)
. |        ⇒
. |        lnk(e) = l
. |        ⇒
. |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. |                                   )))
. |     & (∀e:E.
. |        loc(e) = source(l)
. |        ⇒
. |        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);[<tg,λs,v. f(s(x),v)>])))
. | 19. w : World
. | 20. ∀p:FairFifo.
. |     PossibleWorld(D';w)
. |     ⇒
. |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. |     & (∀e:E.
. |        loc(e) = source(l)
. |        ⇒
. |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. |                                  )))
. |     & (∀e:E.
. |        isrcv(e)
. |        ⇒
. |        lnk(e) = l
. |        ⇒
. |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. |                                  )))
. |     & (∀e:E.
. |        loc(e) = source(l)
. |        ⇒
. |        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);[<tg,λs,v. f(s(x),v)>])))
. | 21. p : FairFifo
. | 22. PossibleWorld(D';w)
. | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | 24. ∀e:E.
. |     loc(e) = source(l)
. |     ⇒
. |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | 25. ∀e:E.
. |     isrcv(e)
. |     ⇒
. |     lnk(e) = l
. |     ⇒
. |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | 26. ∀e:E.
. |     loc(e) = source(l)
. |     ⇒
. |     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);[<tg
. |                                                    ,λs,v. f
. |                                                           (s(x)
. |                                                           ,v)>])
. | 27. vartype(source(l);x) ⊆r A
. | 28. e : E
. | 29. loc(e) = source(l)
. | 30. kind(e) = k
. | 31. valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e))
. | 32. ¬k ∈ dom(k : B)
. | ⊢  B = rcv(l; tg) : T(k)?Top
. |   by Analyze -1
. |      THEN
. |      RWO
. |      Thm* ∀eq:EqDecider(A), x,y:A, v:Top. x ∈ dom(y : v) ⟺ x = y
. |      {fpf_DASH_single_DASH_dom}
. |      0
. |
. |\
. | 1. x : Id
. | 2. tg : Id
. | 3. k : Knd
. | 4. l : IdLnk
. | 5. T : Type
. | 6. A : Type
. | 7. B : Type
. | 8. f : A→B→(T List)
. | 9. rcv(l; tg) = k ⇒ T = B
. | 10. @source(l): ma-single-sends(x : A;
. | |                              k : B ⊕ rcv(l; tg) : T;
. | |                              k;
. | |                              l;
. | |                              [<tg,λs,v. f(s(x),v)>])
. | |   ∈ Dsys
. | 11. ∀D:Dsys.
. | |   @source(l): ma-single-sends(x : A;
. | |                              k : B ⊕ rcv(l; tg) : T;
. | |                              k;
. | |                              l;
. | |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | |   ⇒
. | |   D
. | |   realizes es.(∀x@0:Id.
. | |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |                & (∀e:E.
. | |                   loc(e) = source(l)
. | |                   ⇒
. | |                   (valtype(e) ⊆r ma-valtype
. | |                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | |                & (∀e:E.
. | |                   isrcv(e)
```

```
. | |                  ⇒
. | |                  lnk(e) = l
. | |                  ⇒
. | |                  (valtype(e) ⊆r ma-valtype
. | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | |               & (∀e:E.
. | |                  loc(e) = source(l)
. | |                  ⇒
. | |                  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);[<tg,λs,v. f(s(x),v)>])))
. | 12. @source(l): ma-single-sends(x : A;
. | |                                k : B ⊕ rcv(l; tg) : T;
. | |                                k;
. | |                                l;
. | |                                [<tg,λs,v. f(s(x),v)>])
. | |   ∈ Dsys
. | 13. D : Dsys
. | 14. @source(l): ma-single-sends(x : A;
. | |                                k : B ⊕ rcv(l; tg) : T;
. | |                                k;
. | |                                l;
. | |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | 15. ∀D':Dsys.
. | |   D ⊆ D'
. | |   ⇒
. | |   (∀w:World, p:FairFifo.
. | |    PossibleWorld(D';w)
. | |    ⇒
. | |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |    & (∀e:E.
. | |       loc(e) = source(l)
. | |       ⇒
. | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                  )))
. | |    & (∀e:E.
. | |       isrcv(e)
. | |       ⇒
. | |       lnk(e) = l
. | |       ⇒
. | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                  )))
. | |    & (∀e:E.
. | |       loc(e) = source(l)
. | |       ⇒
. | |       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);[<tg,λs,v. f(s(x),v)>])))))
. | 16. D' : Dsys
. | 17. D ⊆ D'
. | 18. ∀w:World, p:FairFifo.
. | |   PossibleWorld(D';w)
. | |   ⇒
. | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |   & (∀e:E.
. | |      loc(e) = source(l)
. | |      ⇒
. | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                )))
. | |   & (∀e:E.
. | |      isrcv(e)
. | |      ⇒
. | |      lnk(e) = l
. | |      ⇒
. | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                )))
. | |   & (∀e:E.
. | |      loc(e) = source(l)
. | |      ⇒
. | |      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);[<tg,λs,v. f(s(x),v)>])))
. | 19. w : World
. | 20. ∀p:FairFifo.
. | |   PossibleWorld(D';w)
. | |   ⇒
. | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |   & (∀e:E.
. | |      loc(e) = source(l)
. | |      ⇒
. | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                )))
. | |   & (∀e:E.
. | |      isrcv(e)
. | |      ⇒
. | |      lnk(e) = l
. | |      ⇒
```

```
. | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                 )))
. | |   & (∀e:E.
. | |      loc(e) = source(l)
. | |      ⇒
. | |      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);[<tg,λs,v. f(s(x),v)>])))
. | 21. p : FairFifo
. | 22. PossibleWorld(D';w)
. | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | 24. ∀e:E.
. | |   loc(e) = source(l)
. | |   ⇒
. | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | 25. ∀e:E.
. | |   isrcv(e)
. | |   ⇒
. | |   lnk(e) = l
. | |   ⇒
. | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | 26. ∀e:E.
. | |   loc(e) = source(l)
. | |   ⇒
. | |   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);[<tg
. | |                                                 ,λs,v. f
. | |                                                        (s(x)
. | |                                                        ,v)>]))
. | 27. vartype(source(l);x) ⊆r A
. | 28. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | ⊢  ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. | | by AssertBY (k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type) Auto
. | |
. | 1. x : Id
. | 2. tg : Id
. | 3. k : Knd
. | 4. l : IdLnk
. | 5. T : Type
. | 6. A : Type
. | 7. B : Type
. | 8. f : A→B→(T List)
```

```
. | 9. rcv(l; tg) = k ⇒ T = B
. | 10. @source(l): ma-single-sends(x : A;
. | |                              k : B ⊕ rcv(l; tg) : T;
. | |                              k;
. | |                              l;
. | |                              [<tg,λs,v. f(s(x),v)>])
. | |   ∈ Dsys
. | 11. ∀D:Dsys.
. | |   @source(l): ma-single-sends(x : A;
. | |                              k : B ⊕ rcv(l; tg) : T;
. | |                              k;
. | |                              l;
. | |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | |   ⇒
. | |   D
. | |   realizes es.(∀x@0:Id.
. | |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |               & (∀e:E.
. | |                  loc(e) = source(l)
. | |                  ⇒
. | |                  (valtype(e) ⊆r ma-valtype
. | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | |               & (∀e:E.
. | |                  isrcv(e)
. | |                  ⇒
. | |                  lnk(e) = l
. | |                  ⇒
. | |                  (valtype(e) ⊆r ma-valtype
. | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | |               & (∀e:E.
. | |                  loc(e) = source(l)
. | |                  ⇒
. | |                  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);[<tg,λs,v. f(s(x),v)>])))
. | 12. @source(l): ma-single-sends(x : A;
. | |                              k : B ⊕ rcv(l; tg) : T;
. | |                              k;
. | |                              l;
. | |                              [<tg,λs,v. f(s(x),v)>])
. | |   ∈ Dsys
. | 13. D : Dsys
. | 14. @source(l): ma-single-sends(x : A;
. | |                              k : B ⊕ rcv(l; tg) : T;
. | |                              k;
. | |                              l;
```

```
. | |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | 15. ∀D':Dsys.
. | |   D ⊆ D'
. | |   ⇒
. | |   (∀w:World, p:FairFifo.
. | |    PossibleWorld(D';w)
. | |    ⇒
. | |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |    & (∀e:E.
. | |       loc(e) = source(l)
. | |       ⇒
. | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                  )))
. | |    & (∀e:E.
. | |       isrcv(e)
. | |       ⇒
. | |       lnk(e) = l
. | |       ⇒
. | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                  )))
. | |    & (∀e:E.
. | |       loc(e) = source(l)
. | |       ⇒
. | |       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);[<tg,λs,v. f(s(x),v)>])))))
. | 16. D' : Dsys
. | 17. D ⊆ D'
. | 18. ∀w:World, p:FairFifo.
. | |   PossibleWorld(D';w)
. | |   ⇒
. | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |   & (∀e:E.
. | |      loc(e) = source(l)
. | |      ⇒
. | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                 )))
. | |   & (∀e:E.
. | |      isrcv(e)
. | |      ⇒
. | |      lnk(e) = l
. | |      ⇒
. | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                 )))
. | |   & (∀e:E.
. | |      loc(e) = source(l)
. | |      ⇒
. | |      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);[<tg,λs,v. f(s(x),v)>])))
. | 19. w : World
. | 20. ∀p:FairFifo.
. | |   PossibleWorld(D';w)
. | |   ⇒
. | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |   & (∀e:E.
. | |      loc(e) = source(l)
. | |      ⇒
. | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                )))
. | |   & (∀e:E.
. | |      isrcv(e)
. | |      ⇒
. | |      lnk(e) = l
. | |      ⇒
. | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                )))
. | |   & (∀e:E.
. | |      loc(e) = source(l)
. | |      ⇒
. | |      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);[<tg,λs,v. f(s(x),v)>])))
. | 21. p : FairFifo
. | 22. PossibleWorld(D';w)
. | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | 24. ∀e:E.
. | |   loc(e) = source(l)
. | |   ⇒
. | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | 25. ∀e:E.
. | |   isrcv(e)
. | |   ⇒
. | |   lnk(e) = l
. | |   ⇒
. | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | 26. ∀e:E.
. | |   loc(e) = source(l)
. | |   ⇒
```

```
. | |   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);[<tg
. | |                                                 ,λs,v. f
. | |                                                        (s(x)
. | |                                                        ,v)>]))
. | 27. vartype(source(l);x) ⊆r A
. | 28. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | 29. k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type
. | ⊢  ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. | | by ParallelOp -5 THEN ParallelOp -1
. | |
. | |\
. | | .....antecedent .....
. | | 1. x : Id
. | | 2. tg : Id
. | | 3. k : Knd
. | | 4. l : IdLnk
. | | 5. T : Type
. | | 6. A : Type
. | | 7. B : Type
. | | 8. f : A→B→(T List)
. | | 9. rcv(l; tg) = k ⇒ T = B
. | | 10. @source(l): ma-single-sends(x : A;
. | |                                 k : B ⊕ rcv(l; tg) : T;
. | |                                 k;
. | |                                 l;
. | |                                 [<tg,λs,v. f(s(x),v)>])
. | |     ∈ Dsys
. | | 11. ∀D:Dsys.
. | |     @source(l): ma-single-sends(x : A;
. | |                                 k : B ⊕ rcv(l; tg) : T;
. | |                                 k;
. | |                                 l;
. | |                                 [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | |     ⇒
. | |     D
. | |     realizes es.(∀x@0:Id.
. | |                   vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |                  & (∀e:E.
. | |                     loc(e) = source(l)
. | |                     ⇒
. | |                     (valtype(e) ⊆r ma-valtype
. | |                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | |                  & (∀e:E.
. | |                     isrcv(e)
. | |                     ⇒
. | |                     lnk(e) = l
. | |                     ⇒
. | |                     (valtype(e) ⊆r ma-valtype
```

```
. | |                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | |                & (∀e:E.
. | |                   loc(e) = source(l)
. | |                   ⇒
. | |                   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);[<tg
. | |                                         ,λs,v. f(s(x),v)>])))
. | | 12. @source(l): ma-single-sends(x : A;
. | |                                 k : B ⊕ rcv(l; tg) : T;
. | |                                 k;
. | |                                 l;
. | |                                 [<tg,λs,v. f(s(x),v)>])
. | |     ∈ Dsys
. | | 13. D : Dsys
. | | 14. @source(l): ma-single-sends(x : A;
. | |                                 k : B ⊕ rcv(l; tg) : T;
. | |                                 k;
. | |                                 l;
. | |                                 [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | | 15. ∀D':Dsys.
. | |     D ⊆ D'
. | |     ⇒
. | |     (∀w:World, p:FairFifo.
. | |      PossibleWorld(D';w)
. | |      ⇒
. | |      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |      & (∀e:E.
. | |         loc(e) = source(l)
. | |         ⇒
. | |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | |                                   kind(e))))
. | |      & (∀e:E.
. | |         isrcv(e)
. | |         ⇒
. | |         lnk(e) = l
. | |         ⇒
. | |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | |                                   kind(e))))
. | |      & (∀e:E.
. | |         loc(e) = source(l)
. | |         ⇒
. | |         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);[<tg,λs,v. f(s(x),v)>])))
. | | 16. D' : Dsys
. | | 17. D ⊆ D'
. | | 18. ∀w:World, p:FairFifo.
. | |     PossibleWorld(D';w)
. | |     ⇒
. | |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |     & (∀e:E.
. | |        loc(e) = source(l)
. | |        ⇒
. | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                  )))
. | |     & (∀e:E.
. | |        isrcv(e)
. | |        ⇒
. | |        lnk(e) = l
. | |        ⇒
. | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                  )))
. | |     & (∀e:E.
. | |        loc(e) = source(l)
. | |        ⇒
. | |        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);[<tg,λs,v. f(s(x),v)>])))
. | | 19. w : World
. | | 20. ∀p:FairFifo.
. | |     PossibleWorld(D';w)
. | |     ⇒
. | |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | |     & (∀e:E.
. | |        loc(e) = source(l)
. | |        ⇒
. | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                  )))
. | |     & (∀e:E.
. | |        isrcv(e)
. | |        ⇒
. | |        lnk(e) = l
. | |        ⇒
. | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | |                                  )))
```

```
. | |     & (∀e:E.
. | |        loc(e) = source(l)
. | |        ⇒
. | |        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);[<tg,λs,v. f(s(x),v)>])))
. | | 21. p : FairFifo
. | | 22. PossibleWorld(D';w)
. | | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | | 24. ∀e:E.
. | |     loc(e) = source(l)
. | |     ⇒
. | |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | | 25. ∀e:E.
. | |     isrcv(e)
. | |     ⇒
. | |     lnk(e) = l
. | |     ⇒
. | |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | | 26. ∀e:E.
. | |     loc(e) = source(l)
. | |     ⇒
. | |     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);[<tg
. | |                                                    ,λs,v. f
. | |                                                           (s(x)
. | |                                                           ,v)>]))
. | | 27. vartype(source(l);x) ⊆r A
. | | 28. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | | 29. k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type
. | | 30. e : E
. | | 31. kind(e) = rcv(l; tg)
. | | ⊢  isrcv(e)
. | |   by Unfold ‘es-isrcv‘ 0 THEN HypSubst' -1 0
. | |      THEN
. | |      Unfolds [‘isrcv‘;‘rcv‘] 0
. | |      THEN
. | |      Reduce 0
. | |
. | .\
. | . 1. x : Id
```

```
. | . 2. tg : Id
. | . 3. k : Knd
. | . 4. l : IdLnk
. | . 5. T : Type
. | . 6. A : Type
. | . 7. B : Type
. | . 8. f : A→B→(T List)
. | . 9. rcv(l; tg) = k ⇒ T = B
. | . 10. @source(l): ma-single-sends(x : A;
. | . |                              k : B ⊕ rcv(l; tg) : T;
. | . |                              k;
. | . |                              l;
. | . |                              [<tg,λs,v. f(s(x),v)>])
. | . |   ∈ Dsys
. | . 11. ∀D:Dsys.
. | . |   @source(l): ma-single-sends(x : A;
. | . |                              k : B ⊕ rcv(l; tg) : T;
. | . |                              k;
. | . |                              l;
. | . |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . |   ⇒
. | . |   D
. | . |   realizes es.(∀x@0:Id.
. | . |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . |               & (∀e:E.
. | . |                  loc(e) = source(l)
. | . |                  ⇒
. | . |                  (valtype(e) ⊆r ma-valtype
. | . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . |               & (∀e:E.
. | . |                  isrcv(e)
. | . |                  ⇒
. | . |                  lnk(e) = l
. | . |                  ⇒
. | . |                  (valtype(e) ⊆r ma-valtype
. | . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . |               & (∀e:E.
. | . |                  loc(e) = source(l)
. | . |                  ⇒
. | . |                  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);[<tg
. | . |                                        ,λs,v. f(s(x),v)>])))
. | . 12. @source(l): ma-single-sends(x : A;
. | . |                              k : B ⊕ rcv(l; tg) : T;
```

```
. | . |                                k;
. | . |                                l;
. | . |                                [<tg,λs,v. f(s(x),v)>])
. | . |   ∈ Dsys
. | . 13. D : Dsys
. | . 14. @source(l): ma-single-sends(x : A;
. | . |                                k : B ⊕ rcv(l; tg) : T;
. | . |                                k;
. | . |                                l;
. | . |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . 15. ∀D':Dsys.
. | . |   D ⊆ D'
. | . |   ⇒
. | . |   (∀w:World, p:FairFifo.
. | . |    PossibleWorld(D';w)
. | . |    ⇒
. | . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . |    & (∀e:E.
. | . |       loc(e) = source(l)
. | . |       ⇒
. | . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . |                                 kind(e))))
. | . |    & (∀e:E.
. | . |       isrcv(e)
. | . |       ⇒
. | . |       lnk(e) = l
. | . |       ⇒
. | . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . |                                 kind(e))))
. | . |    & (∀e:E.
. | . |       loc(e) = source(l)
. | . |       ⇒
. | . |       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);[<tg,λs,v. f(s(x),v)>]))))
. | . 16. D' : Dsys
. | . 17. D ⊆ D'
. | . 18. ∀w:World, p:FairFifo.
. | . |   PossibleWorld(D';w)
. | . |   ⇒
. | . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . |   & (∀e:E.
. | . |      loc(e) = source(l)
. | . |      ⇒
. | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . |                                )))
. | . |   & (∀e:E.
. | . |      isrcv(e)
```

```
. | . |      ⇒
. | . |      lnk(e) = l
. | . |      ⇒
. | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . |                                 )))
. | . |   & (∀e:E.
. | . |      loc(e) = source(l)
. | . |      ⇒
. | . |      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);[<tg,λs,v. f(s(x),v)>])))
. | . 19. w : World
. | . 20. ∀p:FairFifo.
. | . |   PossibleWorld(D';w)
. | . |   ⇒
. | . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . |   & (∀e:E.
. | . |      loc(e) = source(l)
. | . |      ⇒
. | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . |                                 )))
. | . |   & (∀e:E.
. | . |      isrcv(e)
. | . |      ⇒
. | . |      lnk(e) = l
. | . |      ⇒
. | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . |                                 )))
. | . |   & (∀e:E.
. | . |      loc(e) = source(l)
. | . |      ⇒
. | . |      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);[<tg,λs,v. f(s(x),v)>])))
. | . 21. p : FairFifo
. | . 22. PossibleWorld(D';w)
. | . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | . 24. ∀e:E.
. | . |   loc(e) = source(l)
. | . |   ⇒
. | . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
```

```
. | . 25. ∀e:E.
. | . |   isrcv(e)
. | . |   ⇒
. | . |   lnk(e) = l
. | . |   ⇒
. | . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | . 26. ∀e:E.
. | . |   loc(e) = source(l)
. | . |   ⇒
. | . |   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);[<tg
. | . |                                                 ,λs,v. f
. | . |                                                        (s(x)
. | . |                                                        ,v)>]))
. | . 27. vartype(source(l);x) ⊆r A
. | . 28. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | . 29. k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type
. | . 30. e : E
. | . 31. kind(e) = rcv(l; tg)
. | . 32. lnk(e) = l
. | . |   ⇒
. | . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. | . ⊢  valtype(e) ⊆r T  by Analyze -1
. | . |
. | . |\
. | . | .....antecedent .....
. | . | 1. x : Id
. | . | 2. tg : Id
. | . | 3. k : Knd
. | . | 4. l : IdLnk
. | . | 5. T : Type
. | . | 6. A : Type
. | . | 7. B : Type
. | . | 8. f : A→B→(T List)
. | . | 9. rcv(l; tg) = k ⇒ T = B
. | . | 10. @source(l): ma-single-sends(x : A;
. | . |                                 k : B ⊕ rcv(l; tg) : T;
. | . |                                 k;
. | . |                                 l;
. | . |                                 [<tg,λs,v. f(s(x),v)>])
. | . |     ∈ Dsys
. | . | 11. ∀D:Dsys.
. | . |     @source(l): ma-single-sends(x : A;
. | . |                                 k : B ⊕ rcv(l; tg) : T;
. | . |                                 k;
. | . |                                 l;
. | . |                                 [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . |     ⇒
```

```
. | . |     D
. | . |     realizes es.(∀x@0:Id.
. | . |                  vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . |                 & (∀e:E.
. | . |                    loc(e) = source(l)
. | . |                    ⇒
. | . |                    (valtype(e) ⊆r ma-valtype
. | . |                     (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . |                 & (∀e:E.
. | . |                    isrcv(e)
. | . |                    ⇒
. | . |                    lnk(e) = l
. | . |                    ⇒
. | . |                    (valtype(e) ⊆r ma-valtype
. | . |                     (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . |                 & (∀e:E.
. | . |                    loc(e) = source(l)
. | . |                    ⇒
. | . |                    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);[<tg
. | . |                                          ,λs,v. f(s(x),v)>])))
. | . | 12. @source(l): ma-single-sends(x : A;
. | . |                                  k : B ⊕ rcv(l; tg) : T;
. | . |                                  k;
. | . |                                  l;
. | . |                                  [<tg,λs,v. f(s(x),v)>])
. | . |     ∈ Dsys
. | . | 13. D : Dsys
. | . | 14. @source(l): ma-single-sends(x : A;
. | . |                                  k : B ⊕ rcv(l; tg) : T;
. | . |                                  k;
. | . |                                  l;
. | . |                                  [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . | 15. ∀D':Dsys.
. | . |     D ⊆ D'
. | . |     ⇒
. | . |     (∀w:World, p:FairFifo.
. | . |      PossibleWorld(D';w)
. | . |      ⇒
. | . |      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . |      & (∀e:E.
. | . |         loc(e) = source(l)
. | . |         ⇒
. | . |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
```

```
. | . |                                    kind(e))))
. | . |      & (∀e:E.
. | . |         isrcv(e)
. | . |         ⇒
. | . |         lnk(e) = l
. | . |         ⇒
. | . |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . |                                    kind(e))))
. | . |      & (∀e:E.
. | . |         loc(e) = source(l)
. | . |         ⇒
. | . |         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);[<tg,λs,v. f(s(x),v)>])))))
. | . | 16. D' : Dsys
. | . | 17. D ⊆ D'
. | . | 18. ∀w:World, p:FairFifo.
. | . |     PossibleWorld(D';w)
. | . |     ⇒
. | . |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . |     & (∀e:E.
. | . |        loc(e) = source(l)
. | . |        ⇒
. | . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . |                                   kind(e))))
. | . |     & (∀e:E.
. | . |        isrcv(e)
. | . |        ⇒
. | . |        lnk(e) = l
. | . |        ⇒
. | . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . |                                   kind(e))))
. | . |     & (∀e:E.
. | . |        loc(e) = source(l)
. | . |        ⇒
. | . |        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);[<tg,λs,v. f(s(x),v)>])))
. | . | 19. w : World
```

```
. | . | 20. ∀p:FairFifo.
. | . |     PossibleWorld(D';w)
. | . |     ⇒
. | . |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . |     & (∀e:E.
. | . |        loc(e) = source(l)
. | . |        ⇒
. | . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . |                                  kind(e))))
. | . |     & (∀e:E.
. | . |        isrcv(e)
. | . |        ⇒
. | . |        lnk(e) = l
. | . |        ⇒
. | . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . |                                  kind(e))))
. | . |     & (∀e:E.
. | . |        loc(e) = source(l)
. | . |        ⇒
. | . |        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);[<tg,λs,v. f(s(x),v)>])))
. | . | 21. p : FairFifo
. | . | 22. PossibleWorld(D';w)
. | . | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | . | 24. ∀e:E.
. | . |     loc(e) = source(l)
. | . |     ⇒
. | . |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . |                                ))
. | . | 25. ∀e:E.
. | . |     isrcv(e)
. | . |     ⇒
. | . |     lnk(e) = l
. | . |     ⇒
. | . |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . |                                ))
. | . | 26. ∀e:E.
. | . |     loc(e) = source(l)
. | . |     ⇒
. | . |     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);[<tg,λs,v. f(s(x),v)>]))
. | . | 27. vartype(source(l);x) ⊆r A
. | . | 28. ∀e:E.
. | . |     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | . | 29. k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type
. | . | 30. e : E
. | . | 31. kind(e) = rcv(l; tg)
. | . | ⊢  lnk(e) = l
. | . |   by Unfold ‘es-lnk‘ 0 THEN HypSubst’ -1 0
. | . |      THEN
. | . |      Unfolds [‘lnk‘;‘rcv‘] 0
. | . |      THEN
. | . |      Reduce 0
. | . |
. | . .\
. | . . 1. x : Id
. | . . 2. tg : Id
. | . . 3. k : Knd
. | . . 4. l : IdLnk
. | . . 5. T : Type
. | . . 6. A : Type
. | . . 7. B : Type
. | . . 8. f : A→B→(T List)
. | . . 9. rcv(l; tg) = k ⇒ T = B
. | . . 10. @source(l): ma-single-sends(x : A;
. | . . |                              k : B ⊕ rcv(l; tg) : T;
. | . . |                              k;
. | . . |                              l;
. | . . |                              [<tg,λs,v. f(s(x),v)>])
. | . . |   ∈ Dsys
. | . . 11. ∀D:Dsys.
. | . . |   @source(l): ma-single-sends(x : A;
. | . . |                              k : B ⊕ rcv(l; tg) : T;
. | . . |                              k;
. | . . |                              l;
. | . . |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . . |   ⇒
. | . . |   D
. | . . |   realizes es.(∀x@0:Id.
. | . . |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |               & (∀e:E.
. | . . |                  loc(e) = source(l)
. | . . |                  ⇒
. | . . |                  (valtype(e) ⊆r ma-valtype
. | . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . . |               & (∀e:E.
. | . . |                  isrcv(e)
. | . . |                  ⇒
. | . . |                  lnk(e) = l
. | . . |                  ⇒
. | . . |                  (valtype(e) ⊆r ma-valtype
. | . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . . |               & (∀e:E.
. | . . |                  loc(e) = source(l)
```

```
. | . . |                  ⇒
. | . . |                  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);[<tg
. | . . |                                        ,λs,v. f(s(x),v)>]))))
. | . . 12. @source(l): ma-single-sends(x : A;
. | . . |                               k : B ⊕ rcv(l; tg) : T;
. | . . |                               k;
. | . . |                               l;
. | . . |                               [<tg,λs,v. f(s(x),v)>])
. | . . |   ∈ Dsys
. | . . 13. D : Dsys
. | . . 14. @source(l): ma-single-sends(x : A;
. | . . |                               k : B ⊕ rcv(l; tg) : T;
. | . . |                               k;
. | . . |                               l;
. | . . |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . . 15. ∀D':Dsys.
. | . . |   D ⊆ D'
. | . . |   ⇒
. | . . |   (∀w:World, p:FairFifo.
. | . . |    PossibleWorld(D';w)
. | . . |    ⇒
. | . . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |    & (∀e:E.
. | . . |       loc(e) = source(l)
. | . . |       ⇒
. | . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                  kind(e))))
. | . . |    & (∀e:E.
. | . . |       isrcv(e)
. | . . |       ⇒
. | . . |       lnk(e) = l
. | . . |       ⇒
. | . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                  kind(e))))
. | . . |    & (∀e:E.
. | . . |       loc(e) = source(l)
. | . . |       ⇒
. | . . |       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);[<tg,λs,v. f(s(x),v)>])))
. | . . 16. D' : Dsys
. | . . 17. D ⊆ D'
. | . . 18. ∀w:World, p:FairFifo.
. | . . |   PossibleWorld(D';w)
. | . . |   ⇒
. | . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |   & (∀e:E.
. | . . |      loc(e) = source(l)
. | . . |      ⇒
. | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                 kind(e))))
. | . . |   & (∀e:E.
. | . . |      isrcv(e)
. | . . |      ⇒
. | . . |      lnk(e) = l
. | . . |      ⇒
. | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                 kind(e))))
. | . . |   & (∀e:E.
. | . . |      loc(e) = source(l)
. | . . |      ⇒
. | . . |      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);[<tg,λs,v. f(s(x),v)>])))
. | . . 19. w : World
. | . . 20. ∀p:FairFifo.
. | . . |   PossibleWorld(D';w)
. | . . |   ⇒
. | . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |   & (∀e:E.
. | . . |      loc(e) = source(l)
. | . . |      ⇒
. | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                 kind(e))))
. | . . |   & (∀e:E.
. | . . |      isrcv(e)
. | . . |      ⇒
. | . . |      lnk(e) = l
. | . . |      ⇒
. | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                 kind(e))))
. | . . |   & (∀e:E.
```

```
. | . . |      loc(e) = source(l)
. | . . |      ⇒
. | . . |      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);[<tg,λs,v. f(s(x),v)>])))
. | . . 21. p : FairFifo
. | . . 22. PossibleWorld(D';w)
. | . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | . . 24. ∀e:E.
. | . . |   loc(e) = source(l)
. | . . |   ⇒
. | . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . . |                             ))
. | . . 25. ∀e:E.
. | . . |   isrcv(e)
. | . . |   ⇒
. | . . |   lnk(e) = l
. | . . |   ⇒
. | . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . . |                             ))
. | . . 26. ∀e:E.
. | . . |   loc(e) = source(l)
. | . . |   ⇒
. | . . |   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);[<tg,λs,v. f(s(x),v)>]))
. | . . 27. vartype(source(l);x) ⊆r A
. | . . 28. ∀e:E.
. | . . |   loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | . . 29. k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type
. | . . 30. e : E
. | . . 31. kind(e) = rcv(l; tg)
. | . . 32. valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e))
. | . . ⊢  valtype(e) ⊆r T  by NthHypEq -1 THEN Analyze
. | . . |
. | . . .....subterm .....
. | . . 1. x : Id
. | . . 2. tg : Id
. | . . 3. k : Knd
. | . . 4. l : IdLnk
. | . . 5. T : Type
```

```
. | . . 6. A : Type
. | . . 7. B : Type
. | . . 8. f : A→B→(T List)
. | . . 9. rcv(l; tg) = k ⇒ T = B
. | . . 10. @source(l): ma-single-sends(x : A;
. | . . |                              k : B ⊕ rcv(l; tg) : T;
. | . . |                              k;
. | . . |                              l;
. | . . |                              [<tg,λs,v. f(s(x),v)>])
. | . . |   ∈ Dsys
. | . . 11. ∀D:Dsys.
. | . . |   @source(l): ma-single-sends(x : A;
. | . . |                              k : B ⊕ rcv(l; tg) : T;
. | . . |                              k;
. | . . |                              l;
. | . . |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . . |   ⇒
. | . . |   D
. | . . |   realizes es.(∀x@0:Id.
. | . . |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |               & (∀e:E.
. | . . |                  loc(e) = source(l)
. | . . |                  ⇒
. | . . |                  (valtype(e) ⊆r ma-valtype
. | . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . . |               & (∀e:E.
. | . . |                  isrcv(e)
. | . . |                  ⇒
. | . . |                  lnk(e) = l
. | . . |                  ⇒
. | . . |                  (valtype(e) ⊆r ma-valtype
. | . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . . |               & (∀e:E.
. | . . |                  loc(e) = source(l)
. | . . |                  ⇒
. | . . |                  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);[<tg
. | . . |                                        ,λs,v. f(s(x),v)>])))
. | . . 12. @source(l): ma-single-sends(x : A;
. | . . |                              k : B ⊕ rcv(l; tg) : T;
. | . . |                              k;
. | . . |                              l;
. | . . |                              [<tg,λs,v. f(s(x),v)>])
. | . . |   ∈ Dsys
```

```
. | . . 13. D : Dsys
. | . . 14. @source(l): ma-single-sends(x : A;
. | . . |                              k : B ⊕ rcv(l; tg) : T;
. | . . |                              k;
. | . . |                              l;
. | . . |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . . 15. ∀D':Dsys.
. | . . |   D ⊆ D'
. | . . |   ⇒
. | . . |   (∀w:World, p:FairFifo.
. | . . |    PossibleWorld(D';w)
. | . . |    ⇒
. | . . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |    & (∀e:E.
. | . . |       loc(e) = source(l)
. | . . |       ⇒
. | . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                  kind(e))))
. | . . |    & (∀e:E.
. | . . |       isrcv(e)
. | . . |       ⇒
. | . . |       lnk(e) = l
. | . . |       ⇒
. | . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                  kind(e))))
. | . . |    & (∀e:E.
. | . . |       loc(e) = source(l)
. | . . |       ⇒
. | . . |       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);[<tg,λs,v. f(s(x),v)>]))))
. | . . 16. D' : Dsys
. | . . 17. D ⊆ D'
. | . . 18. ∀w:World, p:FairFifo.
. | . . |   PossibleWorld(D';w)
. | . . |   ⇒
. | . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |   & (∀e:E.
. | . . |      loc(e) = source(l)
. | . . |      ⇒
. | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                 kind(e))))
. | . . |   & (∀e:E.
. | . . |      isrcv(e)
. | . . |      ⇒
. | . . |      lnk(e) = l
```

```
. | . . |      ⇒
. | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                kind(e))))
. | . . |   & (∀e:E.
. | . . |      loc(e) = source(l)
. | . . |      ⇒
. | . . |      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);[<tg,λs,v. f(s(x),v)>])))
. | . . 19. w : World
. | . . 20. ∀p:FairFifo.
. | . . |   PossibleWorld(D';w)
. | . . |   ⇒
. | . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |   & (∀e:E.
. | . . |      loc(e) = source(l)
. | . . |      ⇒
. | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                kind(e))))
. | . . |   & (∀e:E.
. | . . |      isrcv(e)
. | . . |      ⇒
. | . . |      lnk(e) = l
. | . . |      ⇒
. | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                kind(e))))
. | . . |   & (∀e:E.
. | . . |      loc(e) = source(l)
. | . . |      ⇒
. | . . |      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);[<tg,λs,v. f(s(x),v)>])))
. | . . 21. p : FairFifo
. | . . 22. PossibleWorld(D';w)
. | . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | . . 24. ∀e:E.
. | . . |   loc(e) = source(l)
. | . . |   ⇒
. | . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . . |                            ))
. | . . 25. ∀e:E.
```

```
. | . . |   isrcv(e)
. | . . |   ⇒
. | . . |   lnk(e) = l
. | . . |   ⇒
. | . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . . |                             ))
. | . . 26. ∀e:E.
. | . . |   loc(e) = source(l)
. | . . |   ⇒
. | . . |   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);[<tg,λs,v. f(s(x),v)>]))
. | . . 27. vartype(source(l);x) ⊆r A
. | . . 28. ∀e:E.
. | . . |   loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | . . 29. k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type
. | . . 30. e : E
. | . . 31. kind(e) = rcv(l; tg)
. | . . 32. valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e))
. | . . ⊢  T = ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e))
. | . . | by Unfold ‘ma-valtype‘ 0
. | . . |
. | . . 1. x : Id
. | . . 2. tg : Id
. | . . 3. k : Knd
. | . . 4. l : IdLnk
. | . . 5. T : Type
. | . . 6. A : Type
. | . . 7. B : Type
. | . . 8. f : A→B→(T List)
. | . . 9. rcv(l; tg) = k ⇒ T = B
. | . . 10. @source(l): ma-single-sends(x : A;
. | . . |                              k : B ⊕ rcv(l; tg) : T;
. | . . |                              k;
. | . . |                              l;
. | . . |                              [<tg,λs,v. f(s(x),v)>])
. | . . |   ∈ Dsys
. | . . 11. ∀D:Dsys.
. | . . |   @source(l): ma-single-sends(x : A;
. | . . |                              k : B ⊕ rcv(l; tg) : T;
. | . . |                              k;
. | . . |                              l;
. | . . |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . . |   ⇒
. | . . |   D
. | . . |   realizes es.(∀x@0:Id.
. | . . |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |               & (∀e:E.
```

```
. | . . |                  loc(e) = source(l)
. | . . |                  ⇒
. | . . |                  (valtype(e) ⊆r ma-valtype
. | . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . . |               & (∀e:E.
. | . . |                  isrcv(e)
. | . . |                  ⇒
. | . . |                  lnk(e) = l
. | . . |                  ⇒
. | . . |                  (valtype(e) ⊆r ma-valtype
. | . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . . |               & (∀e:E.
. | . . |                  loc(e) = source(l)
. | . . |                  ⇒
. | . . |                  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);[<tg
. | . . |                                        ,λs,v. f(s(x),v)>])))
. | . . 12. @source(l): ma-single-sends(x : A;
. | . . |                               k : B ⊕ rcv(l; tg) : T;
. | . . |                               k;
. | . . |                               l;
. | . . |                               [<tg,λs,v. f(s(x),v)>])
. | . . |   ∈ Dsys
. | . . 13. D : Dsys
. | . . 14. @source(l): ma-single-sends(x : A;
. | . . |                               k : B ⊕ rcv(l; tg) : T;
. | . . |                               k;
. | . . |                               l;
. | . . |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . . 15. ∀D':Dsys.
. | . . |   D ⊆ D'
. | . . |   ⇒
. | . . |   (∀w:World, p:FairFifo.
. | . . |    PossibleWorld(D';w)
. | . . |    ⇒
. | . . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |    & (∀e:E.
. | . . |       loc(e) = source(l)
. | . . |       ⇒
. | . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                  kind(e))))
. | . . |    & (∀e:E.
. | . . |       isrcv(e)
. | . . |       ⇒
```

```
. | . . |       lnk(e) = l
. | . . |       ⇒
. | . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                  kind(e))))
. | . . |    & (∀e:E.
. | . . |       loc(e) = source(l)
. | . . |       ⇒
. | . . |       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);[<tg,λs,v. f(s(x),v)>])))))
. | . . 16. D' : Dsys
. | . . 17. D ⊆ D'
. | . . 18. ∀w:World, p:FairFifo.
. | . . |   PossibleWorld(D';w)
. | . . |   ⇒
. | . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |   & (∀e:E.
. | . . |      loc(e) = source(l)
. | . . |      ⇒
. | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                 kind(e))))
. | . . |   & (∀e:E.
. | . . |      isrcv(e)
. | . . |      ⇒
. | . . |      lnk(e) = l
. | . . |      ⇒
. | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                 kind(e))))
. | . . |   & (∀e:E.
. | . . |      loc(e) = source(l)
. | . . |      ⇒
. | . . |      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);[<tg,λs,v. f(s(x),v)>])))
. | . . 19. w : World
. | . . 20. ∀p:FairFifo.
. | . . |   PossibleWorld(D';w)
. | . . |   ⇒
. | . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
```

```
. | . . |   & (∀e:E.
. | . . |      loc(e) = source(l)
. | . . |      ⇒
. | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                kind(e))))
. | . . |   & (∀e:E.
. | . . |      isrcv(e)
. | . . |      ⇒
. | . . |      lnk(e) = l
. | . . |      ⇒
. | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                kind(e))))
. | . . |   & (∀e:E.
. | . . |      loc(e) = source(l)
. | . . |      ⇒
. | . . |      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);[<tg,λs,v. f(s(x),v)>])))
. | . . 21. p : FairFifo
. | . . 22. PossibleWorld(D';w)
. | . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | . . 24. ∀e:E.
. | . . |   loc(e) = source(l)
. | . . |   ⇒
. | . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . . |                             ))
. | . . 25. ∀e:E.
. | . . |   isrcv(e)
. | . . |   ⇒
. | . . |   lnk(e) = l
. | . . |   ⇒
. | . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . . |                             ))
. | . . 26. ∀e:E.
. | . . |   loc(e) = source(l)
. | . . |   ⇒
. | . . |   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);[<tg,λs,v. f(s(x),v)>]))
. | . . 27. vartype(source(l);x) ⊆r A
. | . . 28. ∀e:E.
```

```
. | . . |   loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | . . 29. k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type
. | . . 30. e : E
. | . . 31. kind(e) = rcv(l; tg)
. | . . 32. valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e))
. | . . ⊢  T = k : B ⊕ rcv(l; tg) : T(kind(e))?Top
. | . . | by ReduceFpf 0 THEN Repeat SplitOnConclITE
. | . . |
. | . . |\
. | . . | .....truecase .....
. | . . | 1. x : Id
. | . . | 2. tg : Id
. | . . | 3. k : Knd
. | . . | 4. l : IdLnk
. | . . | 5. T : Type
. | . . | 6. A : Type
. | . . | 7. B : Type
. | . . | 8. f : A→B→(T List)
. | . . | 9. rcv(l; tg) = k ⇒ T = B
. | . . | 10. @source(l): ma-single-sends(x : A;
. | . . |                                 k : B ⊕ rcv(l; tg) : T;
. | . . |                                 k;
. | . . |                                 l;
. | . . |                                 [<tg,λs,v. f(s(x),v)>])
. | . . |     ∈ Dsys
. | . . | 11. ∀D:Dsys.
. | . . |     @source(l): ma-single-sends(x : A;
. | . . |                                 k : B ⊕ rcv(l; tg) : T;
. | . . |                                 k;
. | . . |                                 l;
. | . . |                                 [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . . |     ⇒
. | . . |     D
. | . . |     realizes es.(∀x@0:Id.
. | . . |                  vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |                & (∀e:E.
. | . . |                   loc(e) = source(l)
. | . . |                   ⇒
. | . . |                   (valtype(e) ⊆r ma-valtype
. | . . |                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . . |                & (∀e:E.
. | . . |                   isrcv(e)
. | . . |                   ⇒
. | . . |                   lnk(e) = l
. | . . |                   ⇒
. | . . |                   (valtype(e) ⊆r ma-valtype
. | . . |                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . . |                & (∀e:E.
. | . . |                   loc(e) = source(l)
. | . . |                   ⇒
. | . . |                   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);[<tg
. | . . |                                          ,λs,v. f
. | . . |                                                 (s(x)
. | . . |                                                 ,v)>])))
. | . . | 12. @source(l): ma-single-sends(x : A;
. | . . |                                  k : B ⊕ rcv(l; tg) : T;
. | . . |                                  k;
. | . . |                                  l;
. | . . |                                  [<tg,λs,v. f(s(x),v)>])
. | . . |     ∈ Dsys
. | . . | 13. D : Dsys
. | . . | 14. @source(l): ma-single-sends(x : A;
. | . . |                                  k : B ⊕ rcv(l; tg) : T;
. | . . |                                  k;
. | . . |                                  l;
. | . . |                                  [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . . | 15. ∀D':Dsys.
. | . . |     D ⊆ D'
. | . . |     ⇒
. | . . |     (∀w:World, p:FairFifo.
. | . . |      PossibleWorld(D';w)
. | . . |      ⇒
. | . . |      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |      & (∀e:E.
. | . . |         loc(e) = source(l)
. | . . |         ⇒
. | . . |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                   kind(e))))
. | . . |      & (∀e:E.
. | . . |         isrcv(e)
. | . . |         ⇒
. | . . |         lnk(e) = l
. | . . |         ⇒
. | . . |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                   kind(e))))
. | . . |      & (∀e:E.
. | . . |         loc(e) = source(l)
. | . . |         ⇒
. | . . |         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);[<tg,λs,v. f(s(x),v)>])))
. | . . | 16. D' : Dsys
. | . . | 17. D ⊆ D'
. | . . | 18. ∀w:World, p:FairFifo.
. | . . |     PossibleWorld(D';w)
. | . . |     ⇒
. | . . |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |     & (∀e:E.
. | . . |        loc(e) = source(l)
. | . . |        ⇒
. | . . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                   kind(e))))
. | . . |     & (∀e:E.
. | . . |        isrcv(e)
. | . . |        ⇒
. | . . |        lnk(e) = l
. | . . |        ⇒
. | . . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                   kind(e))))
. | . . |     & (∀e:E.
. | . . |        loc(e) = source(l)
. | . . |        ⇒
. | . . |        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);[<tg,λs,v. f(s(x),v)>])))
. | . . | 19. w : World
. | . . | 20. ∀p:FairFifo.
. | . . |     PossibleWorld(D';w)
. | . . |     ⇒
. | . . |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . |     & (∀e:E.
. | . . |        loc(e) = source(l)
. | . . |        ⇒
. | . . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                   kind(e))))
. | . . |     & (∀e:E.
. | . . |        isrcv(e)
. | . . |        ⇒
. | . . |        lnk(e) = l
. | . . |        ⇒
. | . . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                   kind(e))))
```

```
. | . . |     & (∀e:E.
. | . . |        loc(e) = source(l)
. | . . |        ⇒
. | . . |        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);[<tg,λs,v. f(s(x),v)>])))
. | . . | 21. p : FairFifo
. | . . | 22. PossibleWorld(D';w)
. | . . | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | . . | 24. ∀e:E.
. | . . |     loc(e) = source(l)
. | . . |     ⇒
. | . . |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                kind(e)))
. | . . | 25. ∀e:E.
. | . . |     isrcv(e)
. | . . |     ⇒
. | . . |     lnk(e) = l
. | . . |     ⇒
. | . . |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . |                                kind(e)))
. | . . | 26. ∀e:E.
. | . . |     loc(e) = source(l)
. | . . |     ⇒
. | . . |     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);[<tg,λs,v. f(s(x),v)>]))
. | . . | 27. vartype(source(l);x) ⊆r A
. | . . | 28. ∀e:E.
. | . . |     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | . . | 29. k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type
. | . . | 30. e : E
. | . . | 31. kind(e) = rcv(l; tg)
. | . . | 32. valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . . |                               )
. | . . | 33. k = kind(e)
. | . . | 34. True
. | . . | ⊢  T = B
. | . . |   by BackThruSomeHyp
```

```
. | . . |      THEN
. | . . |      RWO
. | . . |      Thm* ∀a,b:Knd. a = b ⟺ a = b
. | . . |      {assert_DASH_eq_DASH_knd}
. | . . |      -2
. | . . |
. | . . .\
. | . . . .....falsecase .....
. | . . . 1. x : Id
. | . . . 2. tg : Id
. | . . . 3. k : Knd
. | . . . 4. l : IdLnk
. | . . . 5. T : Type
. | . . . 6. A : Type
. | . . . 7. B : Type
. | . . . 8. f : A→B→(T List)
. | . . . 9. rcv(l; tg) = k ⇒ T = B
. | . . . 10. @source(l): ma-single-sends(x : A;
. | . . .                                k : B ⊕ rcv(l; tg) : T;
. | . . .                                k;
. | . . .                                l;
. | . . .                                [<tg,λs,v. f(s(x),v)>])
. | . . .     ∈ Dsys
. | . . . 11. ∀D:Dsys.
. | . . .     @source(l): ma-single-sends(x : A;
. | . . .                                k : B ⊕ rcv(l; tg) : T;
. | . . .                                k;
. | . . .                                l;
. | . . .                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . . .     ⇒
. | . . .     D
. | . . .     realizes es.(∀x@0:Id.
. | . . .                   vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . .                  & (∀e:E.
. | . . .                     loc(e) = source(l)
. | . . .                     ⇒
. | . . .                     (valtype(e) ⊆r ma-valtype
. | . . .                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . . .                  & (∀e:E.
. | . . .                     isrcv(e)
. | . . .                     ⇒
. | . . .                     lnk(e) = l
. | . . .                     ⇒
. | . . .                     (valtype(e) ⊆r ma-valtype
. | . . .                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. | . . .                  & (∀e:E.
. | . . .                     loc(e) = source(l)
. | . . .                     ⇒
. | . . .                     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);[<tg
. | . . .                                          ,λs,v. f
. | . . .                                                 (s(x)
. | . . .                                                 ,v)>])))
. | . . . 12. @source(l): ma-single-sends(x : A;
. | . . .                                 k : B ⊕ rcv(l; tg) : T;
. | . . .                                 k;
. | . . .                                 l;
. | . . .                                 [<tg,λs,v. f(s(x),v)>])
. | . . .     ∈ Dsys
. | . . . 13. D : Dsys
. | . . . 14. @source(l): ma-single-sends(x : A;
. | . . .                                 k : B ⊕ rcv(l; tg) : T;
. | . . .                                 k;
. | . . .                                 l;
. | . . .                                 [<tg,λs,v. f(s(x),v)>]) ⊆ D
. | . . . 15. ∀D':Dsys.
. | . . .     D ⊆ D'
. | . . .     ⇒
. | . . .     (∀w:World, p:FairFifo.
. | . . .      PossibleWorld(D';w)
. | . . .      ⇒
. | . . .      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . .      & (∀e:E.
. | . . .         loc(e) = source(l)
. | . . .         ⇒
. | . . .         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . .                                   kind(e))))
. | . . .      & (∀e:E.
. | . . .         isrcv(e)
. | . . .         ⇒
. | . . .         lnk(e) = l
. | . . .         ⇒
. | . . .         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . .                                   kind(e))))
. | . . .      & (∀e:E.
. | . . .         loc(e) = source(l)
. | . . .         ⇒
. | . . .         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);[<tg,λs,v. f(s(x),v)>])))))
. | . . . 16. D' : Dsys
. | . . . 17. D ⊆ D'
. | . . . 18. ∀w:World, p:FairFifo.
. | . . .     PossibleWorld(D';w)
. | . . .     ⇒
. | . . .     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . .     & (∀e:E.
. | . . .        loc(e) = source(l)
. | . . .        ⇒
. | . . .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . .                                  kind(e))))
. | . . .     & (∀e:E.
. | . . .        isrcv(e)
. | . . .        ⇒
. | . . .        lnk(e) = l
. | . . .        ⇒
. | . . .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . .                                  kind(e))))
. | . . .     & (∀e:E.
. | . . .        loc(e) = source(l)
. | . . .        ⇒
. | . . .        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);[<tg,λs,v. f(s(x),v)>])))
. | . . . 19. w : World
. | . . . 20. ∀p:FairFifo.
. | . . .     PossibleWorld(D';w)
. | . . .     ⇒
. | . . .     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. | . . .     & (∀e:E.
. | . . .        loc(e) = source(l)
. | . . .        ⇒
. | . . .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . .                                  kind(e))))
. | . . .     & (∀e:E.
. | . . .        isrcv(e)
. | . . .        ⇒
. | . . .        lnk(e) = l
. | . . .        ⇒
. | . . .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . .                                  kind(e))))
. | . . .     & (∀e:E.
. | . . .        loc(e) = source(l)
```

```
. | . . .        ⇒
. | . . .        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);[<tg,λs,v. f(s(x),v)>])))
. | . . . 21. p : FairFifo
. | . . . 22. PossibleWorld(D';w)
. | . . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. | . . . 24. ∀e:E.
. | . . .     loc(e) = source(l)
. | . . .     ⇒
. | . . .     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . .                               kind(e)))
. | . . . 25. ∀e:E.
. | . . .     isrcv(e)
. | . . .     ⇒
. | . . .     lnk(e) = l
. | . . .     ⇒
. | . . .     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. | . . .                               kind(e)))
. | . . . 26. ∀e:E.
. | . . .     loc(e) = source(l)
. | . . .     ⇒
. | . . .     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);[<tg,λs,v. f(s(x),v)>]))
. | . . . 27. vartype(source(l);x) ⊆r A
. | . . . 28. ∀e:E.
. | . . .     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | . . . 29. k : B ⊕ rcv(l; tg) : T ∈ k:Knd fp-> Type
. | . . . 30. e : E
. | . . . 31. kind(e) = rcv(l; tg)
. | . . . 32. valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. | . . .                             )
. | . . . 33. ¬k = kind(e)
. | . . . 34. ¬rcv(l; tg) = kind(e)
. | . . . ⊢  T = Top
. | . . .   by Analyze -1
. | . . .      THEN
. | . . .      RWO
```

```
. | . . .      Thm* ∀a,b:Knd. a = b ⟺ a = b
. | . . .      {assert_DASH_eq_DASH_knd}
. | . . .      0
. |
. .\
. . 1. x : Id
. . 2. tg : Id
. . 3. k : Knd
. . 4. l : IdLnk
. . 5. T : Type
. . 6. A : Type
. . 7. B : Type
. . 8. f : A→B→(T List)
. . 9. rcv(l; tg) = k ⇒ T = B
. . 10. @source(l): ma-single-sends(x : A;
. . |                              k : B ⊕ rcv(l; tg) : T;
. . |                              k;
. . |                              l;
. . |                              [<tg,λs,v. f(s(x),v)>])
. . |   ∈ Dsys
. . 11. ∀D:Dsys.
. . |   @source(l): ma-single-sends(x : A;
. . |                              k : B ⊕ rcv(l; tg) : T;
. . |                              k;
. . |                              l;
. . |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . |   ⇒
. . |   D
. . |   realizes es.(∀x@0:Id.
. . |                vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . |               & (∀e:E.
. . |                  loc(e) = source(l)
. . |                  ⇒
. . |                  (valtype(e) ⊆r ma-valtype
. . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . |               & (∀e:E.
. . |                  isrcv(e)
. . |                  ⇒
. . |                  lnk(e) = l
. . |                  ⇒
. . |                  (valtype(e) ⊆r ma-valtype
. . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . |               & (∀e:E.
. . |                  loc(e) = source(l)
. . |                  ⇒
. . |                  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);[<tg,λs,v. f(s(x),v)>])))
. . 12. @source(l): ma-single-sends(x : A;
. . |                               k : B ⊕ rcv(l; tg) : T;
. . |                               k;
. . |                               l;
. . |                               [<tg,λs,v. f(s(x),v)>])
. . |   ∈ Dsys
. . 13. D : Dsys
. . 14. @source(l): ma-single-sends(x : A;
. . |                               k : B ⊕ rcv(l; tg) : T;
. . |                               k;
. . |                               l;
. . |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . 15. ∀D':Dsys.
. . |   D ⊆ D'
. . |   ⇒
. . |   (∀w:World, p:FairFifo.
. . |    PossibleWorld(D';w)
. . |    ⇒
. . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . |    & (∀e:E.
. . |       loc(e) = source(l)
. . |       ⇒
. . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                 )))
. . |    & (∀e:E.
. . |       isrcv(e)
. . |       ⇒
. . |       lnk(e) = l
. . |       ⇒
. . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                 )))
. . |    & (∀e:E.
. . |       loc(e) = source(l)
. . |       ⇒
. . |       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);[<tg,λs,v. f(s(x),v)>]))))
. . 16. D' : Dsys
. . 17. D ⊆ D'
. . 18. ∀w:World, p:FairFifo.
. . |   PossibleWorld(D';w)
. . |   ⇒
. . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . |   & (∀e:E.
. . |      loc(e) = source(l)
. . |      ⇒
```

```
. . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                )))
. . |   & (∀e:E.
. . |      isrcv(e)
. . |      ⇒
. . |      lnk(e) = l
. . |      ⇒
. . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                )))
. . |   & (∀e:E.
. . |      loc(e) = source(l)
. . |      ⇒
. . |      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);[<tg,λs,v. f(s(x),v)>])))
. . 19. w : World
. . 20. ∀p:FairFifo.
. . |   PossibleWorld(D';w)
. . |   ⇒
. . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . |   & (∀e:E.
. . |      loc(e) = source(l)
. . |      ⇒
. . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                )))
. . |   & (∀e:E.
. . |      isrcv(e)
. . |      ⇒
. . |      lnk(e) = l
. . |      ⇒
. . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                )))
. . |   & (∀e:E.
. . |      loc(e) = source(l)
. . |      ⇒
. . |      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);[<tg,λs,v. f(s(x),v)>])))
. . 21. p : FairFifo
. . 22. PossibleWorld(D';w)
. . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
```

```
. . 24. ∀e:E.
. . |   loc(e) = source(l)
. . |   ⇒
. . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . 25. ∀e:E.
. . |   isrcv(e)
. . |   ⇒
. . |   lnk(e) = l
. . |   ⇒
. . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . 26. ∀e:E.
. . |   loc(e) = source(l)
. . |   ⇒
. . |   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);[<tg
. . |                                                  ,λs,v. f
. . |                                                         (s(x)
. . |                                                         ,v)>]))
. . 27. (vartype(source(l);x) ⊆r A)
. . |   & (∀e:E.
. . |      loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B))
. . |   & (∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T))
. . ⊢  ∀e:E.
. . |  loc(e) = source(l)
. . |  ⇒
. . |  kind(e) = k
. . |  ⇒
. . |  (∃L:{e':E| kind(e') = rcv(l; tg) } List.
. . |   (∀e':E. (e' ∈ L) ⟺ kind(e') = rcv(l; tg) & sender(e') = e)
. . |   & (∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2))
. . |   & map(λe'.val(e');L) = f((x when e),val(e)))
. . | by ParallelOp -2 THEN ParallelOp -1 THEN ParallelOp -1
. . |
. . 1. x : Id
. . 2. tg : Id
. . 3. k : Knd
. . 4. l : IdLnk
. . 5. T : Type
. . 6. A : Type
. . 7. B : Type
. . 8. f : A→B→(T List)
. . 9. rcv(l; tg) = k ⇒ T = B
. . 10. @source(l): ma-single-sends(x : A;
. . |                              k : B ⊕ rcv(l; tg) : T;
. . |                              k;
. . |                              l;
. . |                              [<tg,λs,v. f(s(x),v)>])
. . |   ∈ Dsys
. . 11. ∀D:Dsys.
```

```
. . |   @source(l): ma-single-sends(x : A;
. . |                                k : B ⊕ rcv(l; tg) : T;
. . |                                k;
. . |                                l;
. . |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . |   ⇒
. . |   D
. . |   realizes es.(∀x@0:Id.
. . |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . |                & (∀e:E.
. . |                   loc(e) = source(l)
. . |                   ⇒
. . |                   (valtype(e) ⊆r ma-valtype
. . |                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . |                & (∀e:E.
. . |                   isrcv(e)
. . |                   ⇒
. . |                   lnk(e) = l
. . |                   ⇒
. . |                   (valtype(e) ⊆r ma-valtype
. . |                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . |                & (∀e:E.
. . |                   loc(e) = source(l)
. . |                   ⇒
. . |                   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);[<tg,λs,v. f(s(x),v)>])))
. . 12. @source(l): ma-single-sends(x : A;
. . |                                k : B ⊕ rcv(l; tg) : T;
. . |                                k;
. . |                                l;
. . |                                [<tg,λs,v. f(s(x),v)>])
. . |   ∈ Dsys
. . 13. D : Dsys
. . 14. @source(l): ma-single-sends(x : A;
. . |                                k : B ⊕ rcv(l; tg) : T;
. . |                                k;
. . |                                l;
. . |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . 15. ∀D':Dsys.
. . |   D ⊆ D'
. . |   ⇒
. . |   (∀w:World, p:FairFifo.
. . |    PossibleWorld(D';w)
. . |    ⇒
. . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
```

```
. . |    & (∀e:E.
. . |       loc(e) = source(l)
. . |       ⇒
. . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                 )))
. . |    & (∀e:E.
. . |       isrcv(e)
. . |       ⇒
. . |       lnk(e) = l
. . |       ⇒
. . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                 )))
. . |    & (∀e:E.
. . |       loc(e) = source(l)
. . |       ⇒
. . |       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);[<tg,λs,v. f(s(x),v)>])))))
. . 16. D' : Dsys
. . 17. D ⊆ D'
. . 18. ∀w:World, p:FairFifo.
. . |   PossibleWorld(D';w)
. . |   ⇒
. . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . |   & (∀e:E.
. . |      loc(e) = source(l)
. . |      ⇒
. . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                 )))
. . |   & (∀e:E.
. . |      isrcv(e)
. . |      ⇒
. . |      lnk(e) = l
. . |      ⇒
. . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                 )))
. . |   & (∀e:E.
. . |      loc(e) = source(l)
. . |      ⇒
. . |      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);[<tg,λs,v. f(s(x),v)>])))
. . 19. w : World
. . 20. ∀p:FairFifo.
. . |   PossibleWorld(D';w)
. . |   ⇒
. . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . |   & (∀e:E.
. . |      loc(e) = source(l)
. . |      ⇒
. . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                 )))
. . |   & (∀e:E.
. . |      isrcv(e)
. . |      ⇒
. . |      lnk(e) = l
. . |      ⇒
. . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                 )))
. . |   & (∀e:E.
. . |      loc(e) = source(l)
. . |      ⇒
. . |      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);[<tg,λs,v. f(s(x),v)>])))
. . 21. p : FairFifo
. . 22. PossibleWorld(D';w)
. . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . 24. ∀e:E.
. . |   loc(e) = source(l)
. . |   ⇒
. . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . 25. ∀e:E.
. . |   isrcv(e)
. . |   ⇒
. . |   lnk(e) = l
. . |   ⇒
. . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . 26. ∀e:E.
. . |   loc(e) = source(l)
. . |   ⇒
. . |   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);[<tg
```

```
. . |                                                  ,λs,v. f
. . |                                                         (s(x)
. . |                                                         ,v)>]))
. . 27. (vartype(source(l);x) ⊆r A)
. . |   & (∀e:E.
. . |      loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B))
. . |   & (∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T))
. . 28. e : E
. . 29. loc(e) = source(l)
. . 30. kind(e) = k
. . 31. ∃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);[<tg
. . |                                                  ,λs,v. f
. . |                                                         (s(x)
. . |                                                         ,v)>])
. . ⊢  ∃L:{e':E| kind(e') = rcv(l; tg) } List.
. . |  (∀e':E. (e' ∈ L) ⟺ kind(e') = rcv(l; tg) & sender(e') = e)
. . |  & (∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2))
. . |  & map(λe'.val(e');L) = f((x when e),val(e))  by ExRepD
. . |
. . 1. x : Id
. . 2. tg : Id
. . 3. k : Knd
. . 4. l : IdLnk
. . 5. T : Type
. . 6. A : Type
. . 7. B : Type
. . 8. f : A→B→(T List)
. . 9. rcv(l; tg) = k ⇒ T = B
. . 10. @source(l): ma-single-sends(x : A;
. . |                              k : B ⊕ rcv(l; tg) : T;
. . |                              k;
. . |                              l;
. . |                              [<tg,λs,v. f(s(x),v)>])
. . |   ∈ Dsys
. . 11. ∀D:Dsys.
. . |   @source(l): ma-single-sends(x : A;
. . |                              k : B ⊕ rcv(l; tg) : T;
. . |                              k;
. . |                              l;
. . |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . |   ⇒
. . |   D
. . |   realizes es.(∀x@0:Id.
. . |                vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . |               & (∀e:E.
. . |                  loc(e) = source(l)
. . |                  ⇒
. . |                  (valtype(e) ⊆r ma-valtype
. . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . |               & (∀e:E.
```

```
. . |                  isrcv(e)
. . |                  ⇒
. . |                  lnk(e) = l
. . |                  ⇒
. . |                  (valtype(e) ⊆r ma-valtype
. . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . |               & (∀e:E.
. . |                  loc(e) = source(l)
. . |                  ⇒
. . |                  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);[<tg,λs,v. f(s(x),v)>])))
. . 12. @source(l): ma-single-sends(x : A;
. . |                               k : B ⊕ rcv(l; tg) : T;
. . |                               k;
. . |                               l;
. . |                               [<tg,λs,v. f(s(x),v)>])
. . |   ∈ Dsys
. . 13. D : Dsys
. . 14. @source(l): ma-single-sends(x : A;
. . |                               k : B ⊕ rcv(l; tg) : T;
. . |                               k;
. . |                               l;
. . |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . 15. ∀D':Dsys.
. . |   D ⊆ D'
. . |   ⇒
. . |   (∀w:World, p:FairFifo.
. . |    PossibleWorld(D';w)
. . |    ⇒
. . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . |    & (∀e:E.
. . |       loc(e) = source(l)
. . |       ⇒
. . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                  )))
. . |    & (∀e:E.
. . |       isrcv(e)
. . |       ⇒
. . |       lnk(e) = l
. . |       ⇒
. . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                  )))
. . |    & (∀e:E.
. . |       loc(e) = source(l)
. . |       ⇒
```

```
. . |       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);[<tg,λs,v. f(s(x),v)>])))))
. . 16. D' : Dsys
. . 17. D ⊆ D'
. . 18. ∀w:World, p:FairFifo.
. . |   PossibleWorld(D';w)
. . |   ⇒
. . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . |   & (∀e:E.
. . |      loc(e) = source(l)
. . |      ⇒
. . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                 )))
. . |   & (∀e:E.
. . |      isrcv(e)
. . |      ⇒
. . |      lnk(e) = l
. . |      ⇒
. . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                 )))
. . |   & (∀e:E.
. . |      loc(e) = source(l)
. . |      ⇒
. . |      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);[<tg,λs,v. f(s(x),v)>])))
. . 19. w : World
. . 20. ∀p:FairFifo.
. . |   PossibleWorld(D';w)
. . |   ⇒
. . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . |   & (∀e:E.
. . |      loc(e) = source(l)
. . |      ⇒
. . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                 )))
. . |   & (∀e:E.
. . |      isrcv(e)
. . |      ⇒
. . |      lnk(e) = l
```

```
. . |      ⇒
. . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . |                                )))
. . |   & (∀e:E.
. . |      loc(e) = source(l)
. . |      ⇒
. . |      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);[<tg,λs,v. f(s(x),v)>])))
. . 21. p : FairFifo
. . 22. PossibleWorld(D';w)
. . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . 24. ∀e:E.
. . |   loc(e) = source(l)
. . |   ⇒
. . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . 25. ∀e:E.
. . |   isrcv(e)
. . |   ⇒
. . |   lnk(e) = l
. . |   ⇒
. . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . 26. ∀e:E.
. . |   loc(e) = source(l)
. . |   ⇒
. . |   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);[<tg
. . |                                                  ,λs,v. f
. . |                                                         (s(x)
. . |                                                         ,v)>]))
. . 27. vartype(source(l);x) ⊆r A
. . 28. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . 30. e : E
. . 31. loc(e) = source(l)
. . 32. kind(e) = k
. . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . 34. ∀e':E. (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . 36. map(λe'.<tag(e'),val(e')>;L)
. . |   =
. . |   tagged-list-messages(λz.(z when e);val(e);[<tg
```

```
. . |                                                  ,λs,v. f(s(x),v)>])
. . ⊢  ∃L:{e':E| kind(e') = rcv(l; tg) } List.
. . |  (∀e':E. (e' ∈ L) ⟺ kind(e') = rcv(l; tg) & sender(e') = e)
. . |  & (∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2))
. . |  & map(λe'.val(e');L) = f((x when e),val(e))
. . | by Assert (∀e'∈L.kind(e') = rcv(l; tg))
. . |
. . |\
. . | .....assertion .....
. . | 1. x : Id
. . | 2. tg : Id
. . | 3. k : Knd
. . | 4. l : IdLnk
. . | 5. T : Type
. . | 6. A : Type
. . | 7. B : Type
. . | 8. f : A→B→(T List)
. . | 9. rcv(l; tg) = k ⇒ T = B
. . | 10. @source(l): ma-single-sends(x : A;
. . | |                               k : B ⊕ rcv(l; tg) : T;
. . | |                               k;
. . | |                               l;
. . | |                               [<tg,λs,v. f(s(x),v)>])
. . | |   ∈ Dsys
. . | 11. ∀D:Dsys.
. . | |   @source(l): ma-single-sends(x : A;
. . | |                               k : B ⊕ rcv(l; tg) : T;
. . | |                               k;
. . | |                               l;
. . | |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | |   ⇒
. . | |   D
. . | |   realizes es.(∀x@0:Id.
. . | |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |               & (∀e:E.
. . | |                  loc(e) = source(l)
. . | |                  ⇒
. . | |                  (valtype(e) ⊆r ma-valtype
. . | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | |               & (∀e:E.
. . | |                  isrcv(e)
. . | |                  ⇒
. . | |                  lnk(e) = l
. . | |                  ⇒
. . | |                  (valtype(e) ⊆r ma-valtype
. . | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | |               & (∀e:E.
. . | |                  loc(e) = source(l)
. . | |                  ⇒
. . | |                  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);[<tg
. . | |                                        ,λs,v. f(s(x),v)>])))
. . | 12. @source(l): ma-single-sends(x : A;
. . | |                                k : B ⊕ rcv(l; tg) : T;
. . | |                                k;
. . | |                                l;
. . | |                                [<tg,λs,v. f(s(x),v)>])
. . | |   ∈ Dsys
. . | 13. D : Dsys
. . | 14. @source(l): ma-single-sends(x : A;
. . | |                                k : B ⊕ rcv(l; tg) : T;
. . | |                                k;
. . | |                                l;
. . | |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | 15. ∀D':Dsys.
. . | |   D ⊆ D'
. . | |   ⇒
. . | |   (∀w:World, p:FairFifo.
. . | |    PossibleWorld(D';w)
. . | |    ⇒
. . | |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |    & (∀e:E.
. . | |       loc(e) = source(l)
. . | |       ⇒
. . | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                  kind(e))))
. . | |    & (∀e:E.
. . | |       isrcv(e)
. . | |       ⇒
. . | |       lnk(e) = l
. . | |       ⇒
. . | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                  kind(e))))
. . | |    & (∀e:E.
. . | |       loc(e) = source(l)
. . | |       ⇒
. . | |       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);[<tg,λs,v. f(s(x),v)>]))))
. . | 16. D' : Dsys
. . | 17. D ⊆ D'
. . | 18. ∀w:World, p:FairFifo.
```

```
. . | |   PossibleWorld(D';w)
. . | |   ⇒
. . | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                 )))
. . | |   & (∀e:E.
. . | |      isrcv(e)
. . | |      ⇒
. . | |      lnk(e) = l
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                 )))
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      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);[<tg,λs,v. f(s(x),v)>])))
. . | 19. w : World
. . | 20. ∀p:FairFifo.
. . | |   PossibleWorld(D';w)
. . | |   ⇒
. . | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                 )))
. . | |   & (∀e:E.
. . | |      isrcv(e)
. . | |      ⇒
. . | |      lnk(e) = l
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                 )))
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      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);[<tg,λs,v. f(s(x),v)>])))
. . | 21. p : FairFifo
. . | 22. PossibleWorld(D';w)
. . | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . | 24. ∀e:E.
. . | |   loc(e) = source(l)
. . | |   ⇒
. . | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . | 25. ∀e:E.
. . | |   isrcv(e)
. . | |   ⇒
. . | |   lnk(e) = l
. . | |   ⇒
. . | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . | 26. ∀e:E.
. . | |   loc(e) = source(l)
. . | |   ⇒
. . | |   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);[<tg
. . | |                                                ,λs,v. f
. . | |                                                       (s(x)
. . | |                                                       ,v)>]))
. . | 27. vartype(source(l);x) ⊆r A
. . | 28. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . | 30. e : E
. . | 31. loc(e) = source(l)
. . | 32. kind(e) = k
. . | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . | 34. ∀e':E. (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . | 36. map(λe'.<tag(e'),val(e')>;L)
. . | |   =
. . | |   tagged-list-messages(λz.(z when e);val(e);[<tg
. . | |                                              ,λs,v. f
. . | |                                                     (s(x)
. . | |                                                     ,v)>])
. . | ⊢  (∀e'∈L.kind(e') = rcv(l; tg))
. . | | by Unfold `l_all` 0 THEN Parallel0p -3 THEN Analyze -1
. . | |    THEN
. . | |    Parallel0p -2
. . | |    THEN
. . | |    ExRepD
. . | |    THEN
. . | |    BackThru
. . | |    Thm* ∀es:ES, l:IdLnk, tg:Id, e:E.
```

```
. . | |         isrcv(e)
. . | |         ⇒
. . | |         lnk(e) = l ⇒ tag(e) = tg ⇒ kind(e) = rcv(l; tg)
. . | |    {es_DASH_rcv_DASH_kind}
. . | |
. . | 1. x : Id
. . | 2. tg : Id
. . | 3. k : Knd
. . | 4. l : IdLnk
. . | 5. T : Type
. . | 6. A : Type
. . | 7. B : Type
. . | 8. f : A→B→(T List)
. . | 9. rcv(l; tg) = k ⇒ T = B
. . | 10. @source(l): ma-single-sends(x : A;
. . | |                               k : B ⊕ rcv(l; tg) : T;
. . | |                               k;
. . | |                               l;
. . | |                               [<tg,λs,v. f(s(x),v)>])
. . | |   ∈ Dsys
. . | 11. ∀D:Dsys.
. . | |   @source(l): ma-single-sends(x : A;
. . | |                               k : B ⊕ rcv(l; tg) : T;
. . | |                               k;
. . | |                               l;
. . | |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | |   ⇒
. . | |   D
. . | |   realizes es.(∀x@0:Id.
. . | |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |               & (∀e:E.
. . | |                  loc(e) = source(l)
. . | |                  ⇒
. . | |                  (valtype(e) ⊆r ma-valtype
. . | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | |               & (∀e:E.
. . | |                  isrcv(e)
. . | |                  ⇒
. . | |                  lnk(e) = l
. . | |                  ⇒
. . | |                  (valtype(e) ⊆r ma-valtype
. . | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | |               & (∀e:E.
. . | |                  loc(e) = source(l)
. . | |                  ⇒
. . | |                  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);[<tg
. . | |                                       ,λs,v. f(s(x),v)>])))
. . | 12. @source(l): ma-single-sends(x : A;
. . | |                                k : B ⊕ rcv(l; tg) : T;
. . | |                                k;
. . | |                                l;
. . | |                                [<tg,λs,v. f(s(x),v)>])
. . | |   ∈ Dsys
. . | 13. D : Dsys
. . | 14. @source(l): ma-single-sends(x : A;
. . | |                                k : B ⊕ rcv(l; tg) : T;
. . | |                                k;
. . | |                                l;
. . | |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | 15. ∀D':Dsys.
. . | |   D ⊆ D'
. . | |   ⇒
. . | |   (∀w:World, p:FairFifo.
. . | |    PossibleWorld(D';w)
. . | |    ⇒
. . | |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |    & (∀e:E.
. . | |       loc(e) = source(l)
. . | |       ⇒
. . | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                  kind(e))))
. . | |    & (∀e:E.
. . | |       isrcv(e)
. . | |       ⇒
. . | |       lnk(e) = l
. . | |       ⇒
. . | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                  kind(e))))
. . | |    & (∀e:E.
. . | |       loc(e) = source(l)
. . | |       ⇒
. . | |       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);[<tg,λs,v. f(s(x),v)>])))))
. . | 16. D' : Dsys
. . | 17. D ⊆ D'
. . | 18. ∀w:World, p:FairFifo.
. . | |   PossibleWorld(D';w)
. . | |   ⇒
. . | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |   & (∀e:E.
```

```
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                 )))
. . | |   & (∀e:E.
. . | |      isrcv(e)
. . | |      ⇒
. . | |      lnk(e) = l
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                 )))
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      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);[<tg,λs,v. f(s(x),v)>])))
. . | 19. w : World
. . | 20. ∀p:FairFifo.
. . | |   PossibleWorld(D';w)
. . | |   ⇒
. . | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                 )))
. . | |   & (∀e:E.
. . | |      isrcv(e)
. . | |      ⇒
. . | |      lnk(e) = l
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                 )))
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      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);[<tg,λs,v. f(s(x),v)>])))
. . | 21. p : FairFifo
```

```
. . | 22. PossibleWorld(D';w)
. . | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . | 24. ∀e:E.
. . | |   loc(e) = source(l)
. . | |   ⇒
. . | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . | 25. ∀e:E.
. . | |   isrcv(e)
. . | |   ⇒
. . | |   lnk(e) = l
. . | |   ⇒
. . | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . | 26. ∀e:E.
. . | |   loc(e) = source(l)
. . | |   ⇒
. . | |   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);[<tg
. . | |                                                 ,λs,v. f
. . | |                                                        (s(x)
. . | |                                                        ,v)>]))
. . | 27. vartype(source(l);x) ⊆r A
. . | 28. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . | 30. e : E
. . | 31. loc(e) = source(l)
. . | 32. kind(e) = k
. . | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . | 34. ∀e':E. (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . | 36. map(λe'.<tag(e'),val(e')>;L)
. . | |   =
. . | |   tagged-list-messages(λz.(z when e);val(e);[<tg
. . | |                                               ,λs,v. f
. . | |                                                      (s(x)
. . | |                                                      ,v)>])
. . | 37. e' : E
. . | 38. (e' ∈ L) ⇐ (isrcv(e') & lnk(e') = l & sender(e') = e)
. . | 39. (e' ∈ L)
. . | 40. isrcv(e')
. . | 41. lnk(e') = l
. . | 42. sender(e') = e
. . | ⊢  tag(e') = tg
. . | | by AssertBY (f((x when e),val(e)) ∈ T List) MaAuto
. . | |
. . | 1. x : Id
. . | 2. tg : Id
. . | 3. k : Knd
. . | 4. l : IdLnk
```

```
. . | 5. T : Type
. . | 6. A : Type
. . | 7. B : Type
. . | 8. f : A→B→(T List)
. . | 9. rcv(l; tg) = k ⇒ T = B
. . | 10. @source(l): ma-single-sends(x : A;
. . | |                               k : B ⊕ rcv(l; tg) : T;
. . | |                               k;
. . | |                               l;
. . | |                               [<tg,λs,v. f(s(x),v)>])
. . | |   ∈ Dsys
. . | 11. ∀D:Dsys.
. . | |   @source(l): ma-single-sends(x : A;
. . | |                               k : B ⊕ rcv(l; tg) : T;
. . | |                               k;
. . | |                               l;
. . | |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | |   ⇒
. . | |   D
. . | |   realizes es.(∀x@0:Id.
. . | |                vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |               & (∀e:E.
. . | |                  loc(e) = source(l)
. . | |                  ⇒
. . | |                  (valtype(e) ⊆r ma-valtype
. . | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | |               & (∀e:E.
. . | |                  isrcv(e)
. . | |                  ⇒
. . | |                  lnk(e) = l
. . | |                  ⇒
. . | |                  (valtype(e) ⊆r ma-valtype
. . | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | |               & (∀e:E.
. . | |                  loc(e) = source(l)
. . | |                  ⇒
. . | |                  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);[<tg
. . | |                                        ,λs,v. f(s(x),v)>])))
. . | 12. @source(l): ma-single-sends(x : A;
. . | |                               k : B ⊕ rcv(l; tg) : T;
. . | |                               k;
. . | |                               l;
. . | |                               [<tg,λs,v. f(s(x),v)>])
```

```
. . | |   ∈ Dsys
. . | 13. D : Dsys
. . | 14. @source(l): ma-single-sends(x : A;
. . | |                                k : B ⊕ rcv(l; tg) : T;
. . | |                                k;
. . | |                                l;
. . | |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | 15. ∀D':Dsys.
. . | |   D ⊆ D'
. . | |   ⇒
. . | |   (∀w:World, p:FairFifo.
. . | |    PossibleWorld(D';w)
. . | |    ⇒
. . | |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |    & (∀e:E.
. . | |       loc(e) = source(l)
. . | |       ⇒
. . | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                  kind(e))))
. . | |    & (∀e:E.
. . | |       isrcv(e)
. . | |       ⇒
. . | |       lnk(e) = l
. . | |       ⇒
. . | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                  kind(e))))
. . | |    & (∀e:E.
. . | |       loc(e) = source(l)
. . | |       ⇒
. . | |       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);[<tg,λs,v. f(s(x),v)>])))))
. . | 16. D' : Dsys
. . | 17. D ⊆ D'
. . | 18. ∀w:World, p:FairFifo.

. . | |   PossibleWorld(D';w)
. . | |   ⇒
. . | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                 )))
. . | |   & (∀e:E.
. . | |      isrcv(e)
. . | |      ⇒
. . | |      lnk(e) = l
. . | |      ⇒
```

```
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                 )))
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      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);[<tg,λs,v. f(s(x),v)>])))
. . | 19. w : World
. . | 20. ∀p:FairFifo.
. . | |   PossibleWorld(D';w)
. . | |   ⇒
. . | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                 )))
. . | |   & (∀e:E.
. . | |      isrcv(e)
. . | |      ⇒
. . | |      lnk(e) = l
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                 )))
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      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);[<tg,λs,v. f(s(x),v)>])))
. . | 21. p : FairFifo
. . | 22. PossibleWorld(D';w)
. . | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . | 24. ∀e:E.
. . | |   loc(e) = source(l)
. . | |   ⇒
. . | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . | 25. ∀e:E.
. . | |   isrcv(e)
. . | |   ⇒
```

```
. . | |   lnk(e) = l
. . | |   ⇒
. . | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . | 26. ∀e:E.
. . | |   loc(e) = source(l)
. . | |   ⇒
. . | |   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);[<tg
. . | |                                                 ,λs,v. f
. . | |                                                        (s(x)
. . | |                                                        ,v)>]))
. . | 27. vartype(source(l);x) ⊆r A
. . | 28. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . | 30. e : E
. . | 31. loc(e) = source(l)
. . | 32. kind(e) = k
. . | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . | 34. ∀e':E. (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . | 36. map(λe'.<tag(e'),val(e')>;L)
. . | |   =
. . | |   tagged-list-messages(λz.(z when e);val(e);[<tg
. . | |                                               ,λs,v. f
. . | |                                                      (s(x)
. . | |                                                      ,v)>])
. . | 37. e' : E
. . | 38. (e' ∈ L) ⇐ (isrcv(e') & lnk(e') = l & sender(e') = e)
. . | 39. (e' ∈ L)
. . | 40. isrcv(e')
. . | 41. lnk(e') = l
. . | 42. sender(e') = e
. . | 43. f((x when e),val(e)) ∈ T List
. . | ⊢  tag(e') = tg
. . | | by Unfold `tagged-list-messages` -8 THEN Unfold `concat` -8
. . | |    THEN
. . | |    Reduce -8
. . | |    THEN
. . | |    RWO Thm* ∀l:T List. (l @ nil) ∼ l    {append_nil_sq} -8
. . | |
. . | 1. x : Id
. . | 2. tg : Id
. . | 3. k : Knd
. . | 4. l : IdLnk
. . | 5. T : Type
. . | 6. A : Type
. . | 7. B : Type
. . | 8. f : A→B→(T List)
```

```
. . | 9. rcv(l; tg) = k ⇒ T = B
. . | 10. @source(l): ma-single-sends(x : A;
. . | |                              k : B ⊕ rcv(l; tg) : T;
. . | |                              k;
. . | |                              l;
. . | |                              [<tg,λs,v. f(s(x),v)>])
. . | |   ∈ Dsys
. . | 11. ∀D:Dsys.
. . | |   @source(l): ma-single-sends(x : A;
. . | |                              k : B ⊕ rcv(l; tg) : T;
. . | |                              k;
. . | |                              l;
. . | |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | |   ⇒
. . | |   D
. . | |   realizes es.(∀x@0:Id.
. . | |                vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |               & (∀e:E.
. . | |                  loc(e) = source(l)
. . | |                  ⇒
. . | |                  (valtype(e) ⊆r ma-valtype
. . | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | |               & (∀e:E.
. . | |                  isrcv(e)
. . | |                  ⇒
. . | |                  lnk(e) = l
. . | |                  ⇒
. . | |                  (valtype(e) ⊆r ma-valtype
. . | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | |               & (∀e:E.
. . | |                  loc(e) = source(l)
. . | |                  ⇒
. . | |                  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);[<tg
. . | |                                        ,λs,v. f(s(x),v)>])))
. . | 12. @source(l): ma-single-sends(x : A;
. . | |                              k : B ⊕ rcv(l; tg) : T;
. . | |                              k;
. . | |                              l;
. . | |                              [<tg,λs,v. f(s(x),v)>])
. . | |   ∈ Dsys
. . | 13. D : Dsys
. . | 14. @source(l): ma-single-sends(x : A;
. . | |                              k : B ⊕ rcv(l; tg) : T;
```

```
. . | |                                k;
. . | |                                l;
. . | |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | 15. ∀D':Dsys.
. . | |   D ⊆ D'
. . | |   ⇒
. . | |   (∀w:World, p:FairFifo.
. . | |    PossibleWorld(D';w)
. . | |    ⇒
. . | |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |    & (∀e:E.
. . | |       loc(e) = source(l)
. . | |       ⇒
. . | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                  kind(e))))
. . | |    & (∀e:E.
. . | |       isrcv(e)
. . | |       ⇒
. . | |       lnk(e) = l
. . | |       ⇒
. . | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                  kind(e))))
. . | |    & (∀e:E.
. . | |       loc(e) = source(l)
. . | |       ⇒
. . | |       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);[<tg,λs,v. f(s(x),v)>])))
. . | 16. D' : Dsys
. . | 17. D ⊆ D'
. . | 18. ∀w:World, p:FairFifo.
. . | |   PossibleWorld(D';w)
. . | |   ⇒
. . | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                  )))
. . | |   & (∀e:E.
. . | |      isrcv(e)
. . | |      ⇒
. . | |      lnk(e) = l
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                  )))
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
```

```
. . | |      ⇒
. . | |      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);[<tg,λs,v. f(s(x),v)>])))
. . | 19. w : World
. . | 20. ∀p:FairFifo.
. . | |   PossibleWorld(D';w)
. . | |   ⇒
. . | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                )))
. . | |   & (∀e:E.
. . | |      isrcv(e)
. . | |      ⇒
. . | |      lnk(e) = l
. . | |      ⇒
. . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                )))
. . | |   & (∀e:E.
. . | |      loc(e) = source(l)
. . | |      ⇒
. . | |      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);[<tg,λs,v. f(s(x),v)>])))
. . | 21. p : FairFifo
. . | 22. PossibleWorld(D';w)
. . | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . | 24. ∀e:E.
. . | |   loc(e) = source(l)
. . | |   ⇒
. . | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . | 25. ∀e:E.
. . | |   isrcv(e)
. . | |   ⇒
. . | |   lnk(e) = l
. . | |   ⇒
. . | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . | 26. ∀e:E.
```

```
. . | |   loc(e) = source(l)
. . | |   ⇒
. . | |   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);[<tg
. . | |                                                 ,λs,v. f
. . | |                                                        (s(x)
. . | |                                                        ,v)>]))
. . | 27. vartype(source(l);x) ⊆r A
. . | 28. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . | 30. e : E
. . | 31. loc(e) = source(l)
. . | 32. kind(e) = k
. . | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . | 34. ∀e':E. (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . | 36. map(λe'.<tag(e'),val(e')>;L)
. . | |   =
. . | |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . | 37. e' : E
. . | 38. (e' ∈ L) ⇐ (isrcv(e') & lnk(e') = l & sender(e') = e)
. . | 39. (e' ∈ L)
. . | 40. isrcv(e')
. . | 41. lnk(e') = l
. . | 42. sender(e') = e
. . | 43. f((x when e),val(e)) ∈ T List
. . | ⊢  tag(e') = tg
. . | | by Analyze -5 THEN ExRepD
. . | |    THEN
. . | |    AssertBY
. . | |    (map(λe'.<tag(e'),val(e')>;L)[i]
. . | |     =
. . | |     map(λx@0.<tg,x@0>;f((x when e),val(e)))[i])
. . | |    (Analyze
. . | |     THEN
. . | |     RWO
. . | |     Thm* ∀f:Top, T:Type, L:T List. ||map(f;L)|| ~ ||L||
. . | |     {length_DASH_map}
. . | |     0)
. . | |    THEN
. . | |    RWO
. . | |    Thm* ∀f:Top, T:Type, L:T List, i:ℕ||L||.
. . | |         map(f;L)[i] ~ (f(L[i]))
. . | |    {select_DASH_map}
. . | |    -1
. . | |    THEN
. . | |    Reduce -1
. . | |
```

```
. . | |\
. . | | 1. x : Id
. . | | 2. tg : Id
. . | | 3. k : Knd
. . | | 4. l : IdLnk
. . | | 5. T : Type
. . | | 6. A : Type
. . | | 7. B : Type
. . | | 8. f : A→B→(T List)
. . | | 9. rcv(l; tg) = k ⇒ T = B
. . | | 10. @source(l): ma-single-sends(x : A;
. . | |                                 k : B ⊕ rcv(l; tg) : T;
. . | |                                 k;
. . | |                                 l;
. . | |                                 [<tg,λs,v. f(s(x),v)>])
. . | |     ∈ Dsys
. . | | 11. ∀D:Dsys.
. . | |     @source(l): ma-single-sends(x : A;
. . | |                                 k : B ⊕ rcv(l; tg) : T;
. . | |                                 k;
. . | |                                 l;
. . | |                                 [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | |     ⇒
. . | |     D
. . | |     realizes es.(∀x@0:Id.
. . | |                   vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |                  & (∀e:E.
. . | |                     loc(e) = source(l)
. . | |                     ⇒
. . | |                     (valtype(e) ⊆r ma-valtype
. . | |                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | |                  & (∀e:E.
. . | |                     isrcv(e)
. . | |                     ⇒
. . | |                     lnk(e) = l
. . | |                     ⇒
. . | |                     (valtype(e) ⊆r ma-valtype
. . | |                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | |                  & (∀e:E.
. . | |                     loc(e) = source(l)
. . | |                     ⇒
. . | |                     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);[<tg
. . | |                                           ,λs,v. f(s(x),v)>])))
```

```
. . | | 12. @source(l): ma-single-sends(x : A;
. . | |                                k : B ⊕ rcv(l; tg) : T;
. . | |                                k;
. . | |                                l;
. . | |                                [<tg,λs,v. f(s(x),v)>])
. . | |     ∈ Dsys
. . | | 13. D : Dsys
. . | | 14. @source(l): ma-single-sends(x : A;
. . | |                                k : B ⊕ rcv(l; tg) : T;
. . | |                                k;
. . | |                                l;
. . | |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | | 15. ∀D':Dsys.
. . | |     D ⊆ D'
. . | |     ⇒
. . | |     (∀w:World, p:FairFifo.
. . | |      PossibleWorld(D';w)
. . | |      ⇒
. . | |      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |      & (∀e:E.
. . | |         loc(e) = source(l)
. . | |         ⇒
. . | |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                   kind(e))))
. . | |      & (∀e:E.
. . | |         isrcv(e)
. . | |         ⇒
. . | |         lnk(e) = l
. . | |         ⇒
. . | |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                   kind(e))))
. . | |      & (∀e:E.
. . | |         loc(e) = source(l)
. . | |         ⇒
. . | |         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);[<tg,λs,v. f(s(x),v)>])))
. . | | 16. D' : Dsys
. . | | 17. D ⊆ D'
. . | | 18. ∀w:World, p:FairFifo.
. . | |     PossibleWorld(D';w)
. . | |     ⇒
. . | |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |     & (∀e:E.
. . | |        loc(e) = source(l)
. . | |        ⇒
```

```
. . | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                   kind(e))))
. . | |     & (∀e:E.
. . | |        isrcv(e)
. . | |        ⇒
. . | |        lnk(e) = l
. . | |        ⇒
. . | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                   kind(e))))
. . | |     & (∀e:E.
. . | |        loc(e) = source(l)
. . | |        ⇒
. . | |        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);[<tg,λs,v. f(s(x),v)>])))
. . | | 19. w : World
. . | | 20. ∀p:FairFifo.
. . | |     PossibleWorld(D';w)
. . | |     ⇒
. . | |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | |     & (∀e:E.
. . | |        loc(e) = source(l)
. . | |        ⇒
. . | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                   kind(e))))
. . | |     & (∀e:E.
. . | |        isrcv(e)
. . | |        ⇒
. . | |        lnk(e) = l
. . | |        ⇒
. . | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | |                                   kind(e))))
. . | |     & (∀e:E.
. . | |        loc(e) = source(l)
. . | |        ⇒
. . | |        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);[<tg,λs,v. f(s(x),v)>])))
. . | | 21. p : FairFifo
. . | | 22. PossibleWorld(D';w)
. . | | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
```

```
. . | | 24. ∀e:E.
. . | |     loc(e) = source(l)
. . | |     ⇒
. . | |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                ))
. . | | 25. ∀e:E.
. . | |     isrcv(e)
. . | |     ⇒
. . | |     lnk(e) = l
. . | |     ⇒
. . | |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | |                                ))
. . | | 26. ∀e:E.
. . | |     loc(e) = source(l)
. . | |     ⇒
. . | |     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);[<tg,λs,v. f(s(x),v)>]))
. . | | 27. vartype(source(l);x) ⊆r A
. . | | 28. ∀e:E.
. . | |     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . | | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . | | 30. e : E
. . | | 31. loc(e) = source(l)
. . | | 32. kind(e) = k
. . | | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . | | 34. ∀e':E.
. . | |     (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . | | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . | | 36. map(λe'.<tag(e'),val(e')>;L)
. . | |     =
. . | |     map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . | | 37. e' : E
. . | | 38. (e' ∈ L) ⇐ (isrcv(e') & lnk(e') = l & sender(e') = e)
. . | | 39. i : ℕ
. . | | 40. i<||L||
. . | | 41. e' = L[i]
. . | | 42. isrcv(e')
. . | | 43. lnk(e') = l
. . | | 44. sender(e') = e
. . | | 45. f((x when e),val(e)) ∈ T List
. . | | 46. map(λe'.<tag(e'),val(e')>;L)[i]
. . | |     =
. . | |     map(λx@0.<tg,x@0>;f((x when e),val(e)))[i]
. . | | ⊢  i<||f((x when e),val(e))||
. . | |   by AllHyps
. . | |      (λh.
. . | |       ApFunToHypEquands ‘Z’ ||Z|| ℕ h THENA Complete Auto
```

```
. . | |       THEN
. . | |       RWO
. . | |       Thm* ∀f:Top, T:Type, L:T List. ||map(f;L)|| ∼ ||L||
. . | |       {length_DASH_map}
. . | |       -1
. . | |       THEN
. . | |       Complete Auto)
. . | |
. . | .\
. . | . 1. x : Id
. . | . 2. tg : Id
. . | . 3. k : Knd
. . | . 4. l : IdLnk
. . | . 5. T : Type
. . | . 6. A : Type
. . | . 7. B : Type
. . | . 8. f : A→B→(T List)
. . | . 9. rcv(l; tg) = k ⇒ T = B
. . | . 10. @source(l): ma-single-sends(x : A;
. . | . |                              k : B ⊕ rcv(l; tg) : T;
. . | . |                              k;
. . | . |                              l;
. . | . |                              [<tg,λs,v. f(s(x),v)>])
. . | . |   ∈ Dsys
. . | . 11. ∀D:Dsys.
. . | . |   @source(l): ma-single-sends(x : A;
. . | . |                              k : B ⊕ rcv(l; tg) : T;
. . | . |                              k;
. . | . |                              l;
. . | . |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | . |   ⇒
. . | . |   D
. . | . |   realizes es.(∀x@0:Id.
. . | . |                vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | . |              & (∀e:E.
. . | . |                 loc(e) = source(l)
. . | . |                 ⇒
. . | . |                 (valtype(e) ⊆r ma-valtype
. . | . |                  (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | . |              & (∀e:E.
. . | . |                 isrcv(e)
. . | . |                 ⇒
. . | . |                 lnk(e) = l
. . | . |                 ⇒
. . | . |                 (valtype(e) ⊆r ma-valtype
. . | . |                  (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | . |              & (∀e:E.
. . | . |                 loc(e) = source(l)
. . | . |                 ⇒
. . | . |                 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);[<tg
. . | . |                                        ,λs,v. f(s(x),v)>])))
. . | . 12. @source(l): ma-single-sends(x : A;
. . | . |                               k : B ⊕ rcv(l; tg) : T;
. . | . |                               k;
. . | . |                               l;
. . | . |                               [<tg,λs,v. f(s(x),v)>])
. . | . |   ∈ Dsys
. . | . 13. D : Dsys
. . | . 14. @source(l): ma-single-sends(x : A;
. . | . |                               k : B ⊕ rcv(l; tg) : T;
. . | . |                               k;
. . | . |                               l;
. . | . |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | . 15. ∀D':Dsys.
. . | . |   D ⊆ D'
. . | . |   ⇒
. . | . |   (∀w:World, p:FairFifo.
. . | . |    PossibleWorld(D';w)
. . | . |    ⇒
. . | . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | . |    & (∀e:E.
. . | . |       loc(e) = source(l)
. . | . |       ⇒
. . | . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | . |                                 kind(e))))
. . | . |    & (∀e:E.
. . | . |       isrcv(e)
. . | . |       ⇒
. . | . |       lnk(e) = l
. . | . |       ⇒
. . | . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | . |                                 kind(e))))
. . | . |    & (∀e:E.
. . | . |       loc(e) = source(l)
. . | . |       ⇒
. . | . |       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);[<tg,λs,v. f(s(x),v)>]))))
. . | . 16. D' : Dsys
```

```
. . | . 17. D ⊆ D'
. . | . 18. ∀w:World, p:FairFifo.
. . | . |   PossibleWorld(D';w)
. . | . |   ⇒
. . | . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | . |   & (∀e:E.
. . | . |      loc(e) = source(l)
. . | . |      ⇒
. . | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | . |                                kind(e))))
. . | . |   & (∀e:E.
. . | . |      isrcv(e)
. . | . |      ⇒
. . | . |      lnk(e) = l
. . | . |      ⇒
. . | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | . |                                kind(e))))
. . | . |   & (∀e:E.
. . | . |      loc(e) = source(l)
. . | . |      ⇒
. . | . |      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);[<tg,λs,v. f(s(x),v)>])))
. . | . 19. w : World
. . | . 20. ∀p:FairFifo.
. . | . |   PossibleWorld(D';w)
. . | . |   ⇒
. . | . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | . |   & (∀e:E.
. . | . |      loc(e) = source(l)
. . | . |      ⇒
. . | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | . |                                kind(e))))
. . | . |   & (∀e:E.
. . | . |      isrcv(e)
. . | . |      ⇒
. . | . |      lnk(e) = l
. . | . |      ⇒
. . | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | . |                                kind(e))))
. . | . |   & (∀e:E.
. . | . |      loc(e) = source(l)
. . | . |      ⇒
. . | . |      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);[<tg,λs,v. f(s(x),v)>])))
. . | . 21. p : FairFifo
. . | . 22. PossibleWorld(D';w)
. . | . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . | . 24. ∀e:E.
. . | . |   loc(e) = source(l)
. . | . |   ⇒
. . | . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | . |                             ))
. . | . 25. ∀e:E.
. . | . |   isrcv(e)
. . | . |   ⇒
. . | . |   lnk(e) = l
. . | . |   ⇒
. . | . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | . |                             ))
. . | . 26. ∀e:E.
. . | . |   loc(e) = source(l)
. . | . |   ⇒
. . | . |   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);[<tg,λs,v. f(s(x),v)>]))
. . | . 27. vartype(source(l);x) ⊆r A
. . | . 28. ∀e:E.
. . | . |   loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . | . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . | . 30. e : E
. . | . 31. loc(e) = source(l)
. . | . 32. kind(e) = k
. . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . | . 34. ∀e':E.
. . | . |   (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . | . 36. map(λe'.<tag(e'),val(e')>;L)
. . | . |   =
. . | . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . | . 37. e' : E
. . | . 38. (e' ∈ L) ⇐ (isrcv(e') & lnk(e') = l & sender(e') = e)
. . | . 39. i : ℕ
. . | . 40. i<||L||
. . | . 41. e' = L[i]
. . | . 42. isrcv(e')
. . | . 43. lnk(e') = l
. . | . 44. sender(e') = e
```

```
. . | . 45. f((x when e),val(e)) ∈ T List
. . | . 46. <tag(L[i]),val(L[i])> = <tg,(f((x when e),val(e)))[i]>
. . | . ⊢  tag(e') = tg  by Analyze -1 THEN All Reduce
. . | . |
. . | . 1. x : Id
. . | . 2. tg : Id
. . | . 3. k : Knd
. . | . 4. l : IdLnk
. . | . 5. T : Type
. . | . 6. A : Type
. . | . 7. B : Type
. . | . 8. f : A→B→(T List)
. . | . 9. rcv(l; tg) = k ⇒ T = B
. . | . 10. @source(l): ma-single-sends(x : A;
. . | .                               k : B ⊕ rcv(l; tg) : T;
. . | .                               k;
. . | .                               l;
. . | .                               [<tg,λs,v. f(s(x),v)>])
. . | .     ∈ Dsys
. . | . 11. ∀D:Dsys.
. . | .     @source(l): ma-single-sends(x : A;
. . | .                               k : B ⊕ rcv(l; tg) : T;
. . | .                               k;
. . | .                               l;
. . | .                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | .     ⇒
. . | .     D
. . | .     realizes es.(∀x@0:Id.
. . | .                  vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | .                 & (∀e:E.
. . | .                    loc(e) = source(l)
. . | .                    ⇒
. . | .                    (valtype(e) ⊆r ma-valtype
. . | .                     (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | .                 & (∀e:E.
. . | .                    isrcv(e)
. . | .                    ⇒
. . | .                    lnk(e) = l
. . | .                    ⇒
. . | .                    (valtype(e) ⊆r ma-valtype
. . | .                     (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . | .                 & (∀e:E.
. . | .                    loc(e) = source(l)
. . | .                    ⇒
. . | .                    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);[<tg
. . | .                                          ,λs,v. f(s(x),v)>])))
. . | . 12. @source(l): ma-single-sends(x : A;
. . | .                                 k : B ⊕ rcv(l; tg) : T;
. . | .                                 k;
. . | .                                 l;
. . | .                                 [<tg,λs,v. f(s(x),v)>])
. . | .     ∈ Dsys
. . | . 13. D : Dsys
. . | . 14. @source(l): ma-single-sends(x : A;
. . | .                                 k : B ⊕ rcv(l; tg) : T;
. . | .                                 k;
. . | .                                 l;
. . | .                                 [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . | . 15. ∀D':Dsys.
. . | .     D ⊆ D'
. . | .     ⇒
. . | .     (∀w:World, p:FairFifo.
. . | .      PossibleWorld(D';w)
. . | .      ⇒
. . | .      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | .      & (∀e:E.
. . | .         loc(e) = source(l)
. . | .         ⇒
. . | .         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | .                                   kind(e))))
. . | .      & (∀e:E.
. . | .         isrcv(e)
. . | .         ⇒
. . | .         lnk(e) = l
. . | .         ⇒
. . | .         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | .                                   kind(e))))
. . | .      & (∀e:E.
. . | .         loc(e) = source(l)
. . | .         ⇒
. . | .         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);[<tg,λs,v. f(s(x),v)>]))))
. . | . 16. D' : Dsys
. . | . 17. D ⊆ D'
. . | . 18. ∀w:World, p:FairFifo.
. . | .     PossibleWorld(D';w)
. . | .     ⇒
. . | .     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
```

```
. . | .     & (∀e:E.
. . | .        loc(e) = source(l)
. . | .        ⇒
. . | .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | .                                  kind(e))))
. . | .     & (∀e:E.
. . | .        isrcv(e)
. . | .        ⇒
. . | .        lnk(e) = l
. . | .        ⇒
. . | .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | .                                  kind(e))))
. . | .     & (∀e:E.
. . | .        loc(e) = source(l)
. . | .        ⇒
. . | .        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);[<tg,λs,v. f(s(x),v)>])))
. . | . 19. w : World
. . | . 20. ∀p:FairFifo.
. . | .     PossibleWorld(D';w)
. . | .     ⇒
. . | .     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . | .     & (∀e:E.
. . | .        loc(e) = source(l)
. . | .        ⇒
. . | .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | .                                  kind(e))))
. . | .     & (∀e:E.
. . | .        isrcv(e)
. . | .        ⇒
. . | .        lnk(e) = l
. . | .        ⇒
. . | .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . | .                                  kind(e))))
. . | .     & (∀e:E.
. . | .        loc(e) = source(l)
. . | .        ⇒
. . | .        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);[<tg,λs,v. f(s(x),v)>])))
```

```
. . | . 21. p : FairFifo
. . | . 22. PossibleWorld(D';w)
. . | . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . | . 24. ∀e:E.
. . | .     loc(e) = source(l)
. . | .     ⇒
. . | .     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | .                               ))
. . | . 25. ∀e:E.
. . | .     isrcv(e)
. . | .     ⇒
. . | .     lnk(e) = l
. . | .     ⇒
. . | .     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . | .                               ))
. . | . 26. ∀e:E.
. . | .     loc(e) = source(l)
. . | .     ⇒
. . | .     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);[<tg,λs,v. f(s(x),v)>]))
. . | . 27. vartype(source(l);x) ⊆r A
. . | . 28. ∀e:E.
. . | .     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . | . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . | . 30. e : E
. . | . 31. loc(e) = source(l)
. . | . 32. kind(e) = k
. . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . | . 34. ∀e':E.
. . | .     (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . | . 36. map(λe'.<tag(e'),val(e')>;L)
. . | .     =
. . | .     map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . | . 37. e' : E
. . | . 38. (e' ∈ L) ⇐ (isrcv(e') & lnk(e') = l & sender(e') = e)
. . | . 39. i : ℕ
. . | . 40. i<||L||
. . | . 41. e' = L[i]
. . | . 42. isrcv(e')
. . | . 43. lnk(e') = l
. . | . 44. sender(e') = e
. . | . 45. f((x when e),val(e)) ∈ T List
. . | . 46. tag(L[i]) = tg
. . | . 47. val(L[i]) = (f((x when e),val(e)))[i]
. . | . ⊢  tag(e') = tg  by NthHypEq -2 THEN Analyze THEN Analyze
. . |
```

```
. . .\
. . . 1. x : Id
. . . 2. tg : Id
. . . 3. k : Knd
. . . 4. l : IdLnk
. . . 5. T : Type
. . . 6. A : Type
. . . 7. B : Type
. . . 8. f : A→B→(T List)
. . . 9. rcv(l; tg) = k ⇒ T = B
. . . 10. @source(l): ma-single-sends(x : A;
. . . |                              k : B ⊕ rcv(l; tg) : T;
. . . |                              k;
. . . |                              l;
. . . |                              [<tg,λs,v. f(s(x),v)>])
. . . |   ∈ Dsys
. . . 11. ∀D:Dsys.
. . . |   @source(l): ma-single-sends(x : A;
. . . |                              k : B ⊕ rcv(l; tg) : T;
. . . |                              k;
. . . |                              l;
. . . |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . |   ⇒
. . . |   D
. . . |   realizes es.(∀x@0:Id.
. . . |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . |               & (∀e:E.
. . . |                  loc(e) = source(l)
. . . |                  ⇒
. . . |                  (valtype(e) ⊆r ma-valtype
. . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . |               & (∀e:E.
. . . |                  isrcv(e)
. . . |                  ⇒
. . . |                  lnk(e) = l
. . . |                  ⇒
. . . |                  (valtype(e) ⊆r ma-valtype
. . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . |               & (∀e:E.
. . . |                  loc(e) = source(l)
. . . |                  ⇒
. . . |                  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);[<tg
. . . |                                        ,λs,v. f(s(x),v)>])))
```

```
. . . 12. @source(l): ma-single-sends(x : A;
. . . |                              k : B ⊕ rcv(l; tg) : T;
. . . |                              k;
. . . |                              l;
. . . |                              [<tg,λs,v. f(s(x),v)>])
. . . |   ∈ Dsys
. . . 13. D : Dsys
. . . 14. @source(l): ma-single-sends(x : A;
. . . |                              k : B ⊕ rcv(l; tg) : T;
. . . |                              k;
. . . |                              l;
. . . |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . 15. ∀D':Dsys.
. . . |   D ⊆ D'
. . . |   ⇒
. . . |   (∀w:World, p:FairFifo.
. . . |    PossibleWorld(D';w)
. . . |    ⇒
. . . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . |    & (∀e:E.
. . . |       loc(e) = source(l)
. . . |       ⇒
. . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . |                                  kind(e))))
. . . |    & (∀e:E.
. . . |       isrcv(e)
. . . |       ⇒
. . . |       lnk(e) = l
. . . |       ⇒
. . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . |                                  kind(e))))
. . . |    & (∀e:E.
. . . |       loc(e) = source(l)
. . . |       ⇒
. . . |       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);[<tg,λs,v. f(s(x),v)>])))))
. . . 16. D' : Dsys
. . . 17. D ⊆ D'
. . . 18. ∀w:World, p:FairFifo.
. . . |   PossibleWorld(D';w)
. . . |   ⇒
. . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . |   & (∀e:E.
. . . |      loc(e) = source(l)
. . . |      ⇒
. . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . . |                                  )))
```

```
. . . |   & (∀e:E.
. . . |      isrcv(e)
. . . |      ⇒
. . . |      lnk(e) = l
. . . |      ⇒
. . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . . |                                  )))
. . . |   & (∀e:E.
. . . |      loc(e) = source(l)
. . . |      ⇒
. . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . 19. w : World
. . . 20. ∀p:FairFifo.
. . . |   PossibleWorld(D';w)
. . . |   ⇒
. . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . |   & (∀e:E.
. . . |      loc(e) = source(l)
. . . |      ⇒
. . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . . |                                  )))
. . . |   & (∀e:E.
. . . |      isrcv(e)
. . . |      ⇒
. . . |      lnk(e) = l
. . . |      ⇒
. . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . . |                                  )))
. . . |   & (∀e:E.
. . . |      loc(e) = source(l)
. . . |      ⇒
. . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . 21. p : FairFifo
. . . 22. PossibleWorld(D';w)
. . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . 24. ∀e:E.
. . . |   loc(e) = source(l)
```

```
. . . |   ⇒
. . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . 25. ∀e:E.
. . . |   isrcv(e)
. . . |   ⇒
. . . |   lnk(e) = l
. . . |   ⇒
. . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . 26. ∀e:E.
. . . |   loc(e) = source(l)
. . . |   ⇒
. . . |   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);[<tg
. . . |                                                  ,λs,v. f
. . . |                                                         (s(x)
. . . |                                                         ,v)>]))
. . . 27. vartype(source(l);x) ⊆r A
. . . 28. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . 30. e : E
. . . 31. loc(e) = source(l)
. . . 32. kind(e) = k
. . . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . 34. ∀e':E. (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . 36. map(λe'.<tag(e'),val(e')>;L)
. . . |   =
. . . |   tagged-list-messages(λz.(z when e);val(e);[<tg
. . . |                                                ,λs,v. f
. . . |                                                       (s(x)
. . . |                                                       ,v)>])
. . . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . ⊢  ∃L:{e':E| kind(e') = rcv(l; tg) } List.
. . . |  (∀e':E. (e' ∈ L) ⟺ kind(e') = rcv(l; tg) & sender(e') = e)
. . . |  & (∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2))
. . . |  & map(λe'.val(e');L) = f((x when e),val(e))
. . . | by InstConcl [L]
. . . |
. . . |\
. . . | .....wf .....
. . . | 1. x : Id
. . . | 2. tg : Id
. . . | 3. k : Knd
. . . | 4. l : IdLnk
. . . | 5. T : Type
. . . | 6. A : Type
. . . | 7. B : Type
. . . | 8. f : A→B→(T List)
```

```
. . . | 9. rcv(l; tg) = k ⇒ T = B
. . . | 10. @source(l): ma-single-sends(x : A;
. . . |                                k : B ⊕ rcv(l; tg) : T;
. . . |                                k;
. . . |                                l;
. . . |                                [<tg,λs,v. f(s(x),v)>])
. . . |     ∈ Dsys
. . . | 11. ∀D:Dsys.
. . . |     @source(l): ma-single-sends(x : A;
. . . |                                k : B ⊕ rcv(l; tg) : T;
. . . |                                k;
. . . |                                l;
. . . |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . |     ⇒
. . . |     D
. . . |     realizes es.(∀x@0:Id.
. . . |                  vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . |                 & (∀e:E.
. . . |                    loc(e) = source(l)
. . . |                    ⇒
. . . |                    (valtype(e) ⊆r ma-valtype
. . . |                     (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . |                 & (∀e:E.
. . . |                    isrcv(e)
. . . |                    ⇒
. . . |                    lnk(e) = l
. . . |                    ⇒
. . . |                    (valtype(e) ⊆r ma-valtype
. . . |                     (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . |                 & (∀e:E.
. . . |                    loc(e) = source(l)
. . . |                    ⇒
. . . |                    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);[<tg
. . . |                                          ,λs,v. f(s(x),v)>])))
. . . | 12. @source(l): ma-single-sends(x : A;
. . . |                                k : B ⊕ rcv(l; tg) : T;
. . . |                                k;
. . . |                                l;
. . . |                                [<tg,λs,v. f(s(x),v)>])
. . . |     ∈ Dsys
. . . | 13. D : Dsys
. . . | 14. @source(l): ma-single-sends(x : A;
. . . |                                k : B ⊕ rcv(l; tg) : T;
```

```
. . . |                                  k;
. . . |                                  l;
. . . |                                  [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | 15. ∀D':Dsys.
. . . |     D ⊆ D'
. . . |     ⇒
. . . |     (∀w:World, p:FairFifo.
. . . |      PossibleWorld(D';w)
. . . |      ⇒
. . . |      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . |      & (∀e:E.
. . . |         loc(e) = source(l)
. . . |         ⇒
. . . |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . |                                    kind(e))))
. . . |      & (∀e:E.
. . . |         isrcv(e)
. . . |         ⇒
. . . |         lnk(e) = l
. . . |         ⇒
. . . |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . |                                    kind(e))))
. . . |      & (∀e:E.
. . . |         loc(e) = source(l)
. . . |         ⇒
. . . |         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);[<tg,λs,v. f(s(x),v)>])))))
. . . | 16. D' : Dsys
. . . | 17. D ⊆ D'
. . . | 18. ∀w:World, p:FairFifo.
. . . |     PossibleWorld(D';w)
. . . |     ⇒
. . . |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . |     & (∀e:E.
. . . |        loc(e) = source(l)
. . . |        ⇒
. . . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . |                                    kind(e))))
. . . |     & (∀e:E.
. . . |        isrcv(e)
. . . |        ⇒
. . . |        lnk(e) = l
. . . |        ⇒
. . . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . |                                    kind(e))))
```

```
. . . |     & (∀e:E.
. . . |        loc(e) = source(l)
. . . |        ⇒
. . . |        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);[<tg,λs,v. f(s(x),v)>])))
. . . | 19. w : World
. . . | 20. ∀p:FairFifo.
. . . |     PossibleWorld(D';w)
. . . |     ⇒
. . . |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . |     & (∀e:E.
. . . |        loc(e) = source(l)
. . . |        ⇒
. . . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . |                                   kind(e))))
. . . |     & (∀e:E.
. . . |        isrcv(e)
. . . |        ⇒
. . . |        lnk(e) = l
. . . |        ⇒
. . . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . |                                   kind(e))))
. . . |     & (∀e:E.
. . . |        loc(e) = source(l)
. . . |        ⇒
. . . |        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);[<tg,λs,v. f(s(x),v)>])))
. . . | 21. p : FairFifo
. . . | 22. PossibleWorld(D';w)
. . . | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | 24. ∀e:E.
. . . |     loc(e) = source(l)
. . . |     ⇒
. . . |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . . |                                ))
. . . | 25. ∀e:E.
. . . |     isrcv(e)
. . . |     ⇒
. . . |     lnk(e) = l
```

```
. . . |     ⇒
. . . |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . . |                                ))
. . . | 26. ∀e:E.
. . . |     loc(e) = source(l)
. . . |     ⇒
. . . |     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);[<tg,λs,v. f(s(x),v)>]))
. . . | 27. vartype(source(l);x) ⊆r A
. . . | 28. ∀e:E.
. . . |     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | 30. e : E
. . . | 31. loc(e) = source(l)
. . . | 32. kind(e) = k
. . . | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | 34. ∀e':E.
. . . |     (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | 36. map(λe'.<tag(e'),val(e')>;L)
. . . |     =
. . . |     tagged-list-messages(λz.(z when e);val(e);[<tg
. . . |                                                 ,λs,v. f
. . . |                                                        (s(x)
. . . |                                                        ,v)>])
. . . | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | ⊢  L ∈ {e':E| kind(e') = rcv(l; tg) } List
. . . |   by BackThru
. . . |      Thm* ∀T:Type, L:T List, P:(T→Prop).
. . . |           (∀x∈L.P(x)) ⇒ L ∈ {x:T| P(x) } List
. . . |      {list_DASH_set_DASH_type2}
. . . |
. . . |\
. . . | 1. x : Id
. . . | 2. tg : Id
. . . | 3. k : Knd
. . . | 4. l : IdLnk
. . . | 5. T : Type
. . . | 6. A : Type
. . . | 7. B : Type
. . . | 8. f : A→B→(T List)
. . . | 9. rcv(l; tg) = k ⇒ T = B
. . . | 10. @source(l): ma-single-sends(x : A;
. . . | |                              k : B ⊕ rcv(l; tg) : T;
. . . | |                              k;
. . . | |                              l;
. . . | |                              [<tg,λs,v. f(s(x),v)>])
```

```
. . . | |   ∈ Dsys
. . . | 11. ∀D:Dsys.
. . . | |   @source(l): ma-single-sends(x : A;
. . . | |                               k : B ⊕ rcv(l; tg) : T;
. . . | |                               k;
. . . | |                               l;
. . . | |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | |   ⇒
. . . | |   D
. . . | |   realizes es.(∀x@0:Id.
. . . | |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | |               & (∀e:E.
. . . | |                  loc(e) = source(l)
. . . | |                  ⇒
. . . | |                  (valtype(e) ⊆r ma-valtype
. . . | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | |               & (∀e:E.
. . . | |                  isrcv(e)
. . . | |                  ⇒
. . . | |                  lnk(e) = l
. . . | |                  ⇒
. . . | |                  (valtype(e) ⊆r ma-valtype
. . . | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | |               & (∀e:E.
. . . | |                  loc(e) = source(l)
. . . | |                  ⇒
. . . | |                  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);[<tg
. . . | |                                        ,λs,v. f(s(x),v)>])))
. . . | 12. @source(l): ma-single-sends(x : A;
. . . | |                               k : B ⊕ rcv(l; tg) : T;
. . . | |                               k;
. . . | |                               l;
. . . | |                               [<tg,λs,v. f(s(x),v)>])
. . . | |   ∈ Dsys
. . . | 13. D : Dsys
. . . | 14. @source(l): ma-single-sends(x : A;
. . . | |                               k : B ⊕ rcv(l; tg) : T;
. . . | |                               k;
. . . | |                               l;
. . . | |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | 15. ∀D':Dsys.
. . . | |   D ⊆ D'
. . . | |   ⇒
```

```
. . . | |   (∀w:World, p:FairFifo.
. . . | |    PossibleWorld(D';w)

. . . | |    ⇒
. . . | |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | |    & (∀e:E.
. . . | |       loc(e) = source(l)
. . . | |       ⇒
. . . | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | |                                  kind(e))))
. . . | |    & (∀e:E.
. . . | |       isrcv(e)
. . . | |       ⇒
. . . | |       lnk(e) = l
. . . | |       ⇒
. . . | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | |                                  kind(e))))
. . . | |    & (∀e:E.
. . . | |       loc(e) = source(l)
. . . | |       ⇒
. . . | |       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);[<tg,λs,v. f(s(x),v)>])))))
. . . | 16. D' : Dsys
. . . | 17. D ⊆ D'
. . . | 18. ∀w:World, p:FairFifo.
. . . | |   PossibleWorld(D';w)
. . . | |   ⇒
. . . | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | |   & (∀e:E.
. . . | |      loc(e) = source(l)
. . . | |      ⇒
. . . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | |                                 kind(e))))
. . . | |   & (∀e:E.
. . . | |      isrcv(e)
. . . | |      ⇒
. . . | |      lnk(e) = l
. . . | |      ⇒
. . . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | |                                 kind(e))))
. . . | |   & (∀e:E.
. . . | |      loc(e) = source(l)
. . . | |      ⇒
. . . | |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | 19. w : World
. . . | 20. ∀p:FairFifo.
. . . | |   PossibleWorld(D';w)
. . . | |   ⇒
. . . | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | |   & (∀e:E.
. . . | |      loc(e) = source(l)
. . . | |      ⇒
. . . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | |                                kind(e))))
. . . | |   & (∀e:E.
. . . | |      isrcv(e)
. . . | |      ⇒
. . . | |      lnk(e) = l
. . . | |      ⇒
. . . | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | |                                kind(e))))
. . . | |   & (∀e:E.
. . . | |      loc(e) = source(l)
. . . | |      ⇒
. . . | |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | 21. p : FairFifo
. . . | 22. PossibleWorld(D';w)
. . . | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | 24. ∀e:E.
. . . | |   loc(e) = source(l)
. . . | |   ⇒
. . . | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | |                            ))
. . . | 25. ∀e:E.
. . . | |   isrcv(e)
. . . | |   ⇒
. . . | |   lnk(e) = l
. . . | |   ⇒
. . . | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | |                            ))
. . . | 26. ∀e:E.
. . . | |   loc(e) = source(l)
. . . | |   ⇒
```

```
. . . | |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | 27. vartype(source(l);x) ⊆r A
. . . | 28. ∀e:E.
. . . | |   loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | 30. e : E
. . . | 31. loc(e) = source(l)
. . . | 32. kind(e) = k
. . . | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | 34. ∀e':E.
. . . | |   (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | |   =
. . . | |   tagged-list-messages(λz.(z when e);val(e);[<tg
. . . | |                                              ,λs,v. f
. . . | |                                                     (s(x)
. . . | |                                                     ,v)>])
. . . | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | ⊢  (∀e':E. (e' ∈ L) ⟺ kind(e') = rcv(l; tg) & sender(e') = e)
. . . | |  & (∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2))
. . . | |  & map(λe'.val(e');L) = f((x when e),val(e))
. . . | | by SplitAndConcl THEN Try Trivial
. . . | |
. . . | |\
. . . | | 1. x : Id
. . . | | 2. tg : Id
. . . | | 3. k : Knd
. . . | | 4. l : IdLnk
. . . | | 5. T : Type
. . . | | 6. A : Type
. . . | | 7. B : Type
. . . | | 8. f : A→B→(T List)
. . . | | 9. rcv(l; tg) = k ⇒ T = B
. . . | | 10. @source(l): ma-single-sends(x : A;
. . . | | |                             k : B ⊕ rcv(l; tg) : T;
. . . | | |                             k;
. . . | | |                             l;
. . . | | |                             [<tg,λs,v. f(s(x),v)>])
. . . | | |   ∈ Dsys
. . . | | 11. ∀D:Dsys.
. . . | | |   @source(l): ma-single-sends(x : A;
. . . | | |                             k : B ⊕ rcv(l; tg) : T;
. . . | | |                             k;
. . . | | |                             l;
. . . | | |                             [<tg,λs,v. f(s(x),v)>]) ⊆ D
```

```
. . . | | |   ⇒
. . . | | |   D
. . . | | |   realizes es.(∀x@0:Id.
. . . | | |                vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | |               & (∀e:E.
. . . | | |                  loc(e) = source(l)
. . . | | |                  ⇒
. . . | | |                  (valtype(e) ⊆r ma-valtype
. . . | | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | | |               & (∀e:E.
. . . | | |                  isrcv(e)
. . . | | |                  ⇒
. . . | | |                  lnk(e) = l
. . . | | |                  ⇒
. . . | | |                  (valtype(e) ⊆r ma-valtype
. . . | | |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | | |               & (∀e:E.
. . . | | |                  loc(e) = source(l)
. . . | | |                  ⇒
. . . | | |                  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);[<tg
. . . | | |                                        ,λs,v. f
. . . | | |                                               (s(x)
. . . | | |                                               ,v)>])))
. . . | | 12. @source(l): ma-single-sends(x : A;
. . . | | |                               k : B ⊕ rcv(l; tg) : T;
. . . | | |                               k;
. . . | | |                               l;
. . . | | |                               [<tg,λs,v. f(s(x),v)>])
. . . | | |   ∈ Dsys
. . . | | 13. D : Dsys
. . . | | 14. @source(l): ma-single-sends(x : A;
. . . | | |                               k : B ⊕ rcv(l; tg) : T;
. . . | | |                               k;
. . . | | |                               l;
. . . | | |                               [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | | 15. ∀D':Dsys.
. . . | | |   D ⊆ D'
. . . | | |   ⇒
. . . | | |   (∀w:World, p:FairFifo.
. . . | | |    PossibleWorld(D';w)
```

```
. . . | | |    ⇒
. . . | | |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | |    & (∀e:E.
. . . | | |       loc(e) = source(l)
. . . | | |       ⇒
. . . | | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                  kind(e))))
. . . | | |    & (∀e:E.
. . . | | |       isrcv(e)
. . . | | |       ⇒
. . . | | |       lnk(e) = l
. . . | | |       ⇒
. . . | | |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                  kind(e))))
. . . | | |    & (∀e:E.
. . . | | |       loc(e) = source(l)
. . . | | |       ⇒
. . . | | |       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);[<tg,λs,v. f(s(x),v)>])))))
. . . | | 16. D' : Dsys
. . . | | 17. D ⊆ D'
. . . | | 18. ∀w:World, p:FairFifo.
. . . | | |   PossibleWorld(D';w)
. . . | | |   ⇒
. . . | | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | |   & (∀e:E.
. . . | | |      loc(e) = source(l)
. . . | | |      ⇒
. . . | | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                 kind(e))))
. . . | | |   & (∀e:E.
. . . | | |      isrcv(e)
. . . | | |      ⇒
. . . | | |      lnk(e) = l
. . . | | |      ⇒
. . . | | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                 kind(e))))
. . . | | |   & (∀e:E.
. . . | | |      loc(e) = source(l)
. . . | | |      ⇒
. . . | | |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | | 19. w : World
. . . | | 20. ∀p:FairFifo.
. . . | | |   PossibleWorld(D';w)
. . . | | |   ⇒
. . . | | |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | |   & (∀e:E.
. . . | | |      loc(e) = source(l)
. . . | | |      ⇒
. . . | | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                kind(e))))
. . . | | |   & (∀e:E.
. . . | | |      isrcv(e)
. . . | | |      ⇒
. . . | | |      lnk(e) = l
. . . | | |      ⇒
. . . | | |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                kind(e))))
. . . | | |   & (∀e:E.
. . . | | |      loc(e) = source(l)
. . . | | |      ⇒
. . . | | |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | | 21. p : FairFifo
. . . | | 22. PossibleWorld(D';w)
. . . | | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | | 24. ∀e:E.
. . . | | |   loc(e) = source(l)
. . . | | |   ⇒
. . . | | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                             kind(e)))
. . . | | 25. ∀e:E.
. . . | | |   isrcv(e)
. . . | | |   ⇒
. . . | | |   lnk(e) = l
. . . | | |   ⇒
. . . | | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                             kind(e)))
. . . | | 26. ∀e:E.
```

```
. . . | | |   loc(e) = source(l)
. . . | | |   ⇒
. . . | | |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | | 27. vartype(source(l);x) ⊆r A
. . . | | 28. ∀e:E.
. . . | | |   loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | | 30. e : E
. . . | | 31. loc(e) = source(l)
. . . | | 32. kind(e) = k
. . . | | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | | 34. ∀e':E.
. . . | | |   (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | | |   =
. . . | | |   tagged-list-messages(λz.(z when e);val(e);[<tg
. . . | | |                                              ,λs,v. f
. . . | | |                                                     (s(x)
. . . | | |                                                     ,v)>])
. . . | | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | | ⊢  ∀e':E. (e' ∈ L) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
. . . | | | by ParallelOp -4 THEN ThinTrivial THEN ExRepD
. . . | | |
. . . | | |\
. . . | | | 1. x : Id
. . . | | | 2. tg : Id
. . . | | | 3. k : Knd
. . . | | | 4. l : IdLnk
. . . | | | 5. T : Type
. . . | | | 6. A : Type
. . . | | | 7. B : Type
. . . | | | 8. f : A→B→(T List)
. . . | | | 9. rcv(l; tg) = k ⇒ T = B
. . . | | | 10. @source(l): ma-single-sends(x : A;
. . . | | |                                 k : B ⊕ rcv(l; tg) : T;
. . . | | |                                 k;
. . . | | |                                 l;
. . . | | |                                 [<tg,λs,v. f(s(x),v)>])
. . . | | |     ∈ Dsys
. . . | | | 11. ∀D:Dsys.
. . . | | |     @source(l): ma-single-sends(x : A;
. . . | | |                                 k : B ⊕ rcv(l; tg) : T;
. . . | | |                                 k;
. . . | | |                                 l;
. . . | | |                                 [<tg,λs,v. f(s(x),v)>]
```

```
. . . | | |                                ) ⊆ D
. . . | | |     ⇒
. . . | | |     D
. . . | | |     realizes es.(∀x@0:Id.
. . . | | |                  vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | |                 & (∀e:E.
. . . | | |                    loc(e) = source(l)
. . . | | |                    ⇒
. . . | | |                    (valtype(e) ⊆r ma-valtype
. . . | | |                     (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | | |                 & (∀e:E.
. . . | | |                    isrcv(e)
. . . | | |                    ⇒
. . . | | |                    lnk(e) = l
. . . | | |                    ⇒
. . . | | |                    (valtype(e) ⊆r ma-valtype
. . . | | |                     (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | | |                 & (∀e:E.
. . . | | |                    loc(e) = source(l)
. . . | | |                    ⇒
. . . | | |                    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);[<tg
. . . | | |                                          ,λs,v. f
. . . | | |                                                 (s(x)
. . . | | |                                                 ,v)>])))
. . . | | | 12. @source(l): ma-single-sends(x : A;
. . . | | |                                 k : B ⊕ rcv(l; tg) : T;
. . . | | |                                 k;
. . . | | |                                 l;
. . . | | |                                 [<tg,λs,v. f(s(x),v)>])
. . . | | |     ∈ Dsys
. . . | | | 13. D : Dsys
. . . | | | 14. @source(l): ma-single-sends(x : A;
. . . | | |                                 k : B ⊕ rcv(l; tg) : T;
. . . | | |                                 k;
. . . | | |                                 l;
. . . | | |                                 [<tg,λs,v. f(s(x),v)>]
. . . | | |                                 ) ⊆ D
. . . | | | 15. ∀D':Dsys.
. . . | | |     D ⊆ D'
. . . | | |     ⇒
```

```
. . . | | |     (∀w:World, p:FairFifo.
. . . | | |      PossibleWorld(D';w)
. . . | | |      ⇒
. . . | | |      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | |      & (∀e:E.
. . . | | |         loc(e) = source(l)
. . . | | |         ⇒
. . . | | |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                   kind(e))))
. . . | | |      & (∀e:E.
. . . | | |         isrcv(e)
. . . | | |         ⇒
. . . | | |         lnk(e) = l
. . . | | |         ⇒
. . . | | |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                   kind(e))))
. . . | | |      & (∀e:E.
. . . | | |         loc(e) = source(l)
. . . | | |         ⇒
. . . | | |         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);[<tg,λs,v. f(s(x),v)>])))))
. . . | | | 16. D' : Dsys
. . . | | | 17. D ⊆ D'
. . . | | | 18. ∀w:World, p:FairFifo.
. . . | | |     PossibleWorld(D';w)
. . . | | |     ⇒
. . . | | |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | |     & (∀e:E.
. . . | | |        loc(e) = source(l)
. . . | | |        ⇒
. . . | | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                   kind(e))))
. . . | | |     & (∀e:E.
. . . | | |        isrcv(e)
. . . | | |        ⇒
. . . | | |        lnk(e) = l
. . . | | |        ⇒
. . . | | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                   kind(e))))
. . . | | |     & (∀e:E.
. . . | | |        loc(e) = source(l)
. . . | | |        ⇒
. . . | | |        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);[<tg,λs,v. f(s(x),v)>])))
. . . | | | 19. w : World
. . . | | | 20. ∀p:FairFifo.
. . . | | |     PossibleWorld(D';w)
. . . | | |     ⇒
. . . | | |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | |     & (∀e:E.
. . . | | |        loc(e) = source(l)
. . . | | |        ⇒
. . . | | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                  kind(e))))
. . . | | |     & (∀e:E.
. . . | | |        isrcv(e)
. . . | | |        ⇒
. . . | | |        lnk(e) = l
. . . | | |        ⇒
. . . | | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                  kind(e))))
. . . | | |     & (∀e:E.
. . . | | |        loc(e) = source(l)
. . . | | |        ⇒
. . . | | |        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);[<tg,λs,v. f(s(x),v)>])))
. . . | | | 21. p : FairFifo
. . . | | | 22. PossibleWorld(D';w)
. . . | | | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | | | 24. ∀e:E.
. . . | | |     loc(e) = source(l)
. . . | | |     ⇒
. . . | | |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                kind(e)))
. . . | | | 25. ∀e:E.
. . . | | |     isrcv(e)
. . . | | |     ⇒
. . . | | |     lnk(e) = l
. . . | | |     ⇒
. . . | | |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
```

```
. . . | | |                                kind(e)))
. . . | | | 26. ∀e:E.
. . . | | |     loc(e) = source(l)
. . . | | |     ⇒
. . . | | |     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);[<tg,λs,v. f(s(x),v)>]))
. . . | | | 27. vartype(source(l);x) ⊆r A
. . . | | | 28. ∀e:E.
. . . | | |     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | | | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | | | 30. e : E
. . . | | | 31. loc(e) = source(l)
. . . | | | 32. kind(e) = k
. . . | | | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | | | 34. ∀e':E.
. . . | | |     (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | | | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | | | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | | |     =
. . . | | |     tagged-list-messages(λz.
. . . | | |     (z when e);val(e);[<tg,λs,v. f(s(x),v)>])
. . . | | | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | | | 38. e' : E
. . . | | | 39. (e' ∈ L) ⇐ (isrcv(e') & lnk(e') = l & sender(e') = e)
. . . | | | 40. (e' ∈ L)
. . . | | | 41. isrcv(e')
. . . | | | 42. lnk(e') = l
. . . | | | 43. sender(e') = e
. . . | | | ⊢  kind(e') = rcv(l; tg)
. . . | | |   by AllHyps (λh.Unfold `l_all` h THEN BackThru h)
. . . | | |
. . . | | |\
. . . | | | 1. x : Id
. . . | | | 2. tg : Id
. . . | | | 3. k : Knd
. . . | | | 4. l : IdLnk
. . . | | | 5. T : Type
. . . | | | 6. A : Type
. . . | | | 7. B : Type
. . . | | | 8. f : A→B→(T List)
. . . | | | 9. rcv(l; tg) = k ⇒ T = B
. . . | | | 10. @source(l): ma-single-sends(x : A;
. . . | | |                                 k : B ⊕ rcv(l; tg) : T;
. . . | | |                                 k;
. . . | | |                                 l;
```

```
. . . | | |                                  [<tg,λs,v. f(s(x),v)>])
. . . | | |     ∈ Dsys
. . . | | | 11. ∀D:Dsys.
. . . | | |     @source(l): ma-single-sends(x : A;
. . . | | |                                  k : B ⊕ rcv(l; tg) : T;
. . . | | |                                  k;
. . . | | |                                  l;
. . . | | |                                  [<tg,λs,v. f(s(x),v)>]
. . . | | |                                  ) ⊆ D
. . . | | |     ⇒
. . . | | |     D
. . . | | |     realizes es.(∀x@0:Id.
. . . | | |                   vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | |                  & (∀e:E.
. . . | | |                     loc(e) = source(l)
. . . | | |                     ⇒
. . . | | |                     (valtype(e) ⊆r ma-valtype
. . . | | |                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | | |                  & (∀e:E.
. . . | | |                     isrcv(e)
. . . | | |                     ⇒
. . . | | |                     lnk(e) = l
. . . | | |                     ⇒
. . . | | |                     (valtype(e) ⊆r ma-valtype
. . . | | |                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | | |                  & (∀e:E.
. . . | | |                     loc(e) = source(l)
. . . | | |                     ⇒
. . . | | |                     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);[<tg
. . . | | |                                          ,λs,v. f
. . . | | |                                                 (s(x)
. . . | | |                                                 ,v)>])))
. . . | | | 12. @source(l): ma-single-sends(x : A;
. . . | | |                                  k : B ⊕ rcv(l; tg) : T;
. . . | | |                                  k;
. . . | | |                                  l;
. . . | | |                                  [<tg,λs,v. f(s(x),v)>])
. . . | | |     ∈ Dsys
. . . | | | 13. D : Dsys
. . . | | | 14. @source(l): ma-single-sends(x : A;
```

```
. . . | | |                                  k : B ⊕ rcv(l; tg) : T;
. . . | | |                                  k;
. . . | | |                                  l;
. . . | | |                                  [<tg,λs,v. f(s(x),v)>]
. . . | | |                                  ) ⊆ D
. . . | | | 15. ∀D':Dsys.
. . . | | |     D ⊆ D'
. . . | | |     ⇒
. . . | | |     (∀w:World, p:FairFifo.
. . . | | |      PossibleWorld(D';w)
. . . | | |      ⇒
. . . | | |      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | |      & (∀e:E.
. . . | | |         loc(e) = source(l)
. . . | | |         ⇒
. . . | | |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                   kind(e))))
. . . | | |      & (∀e:E.
. . . | | |         isrcv(e)
. . . | | |         ⇒
. . . | | |         lnk(e) = l
. . . | | |         ⇒
. . . | | |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                   kind(e))))
. . . | | |      & (∀e:E.
. . . | | |         loc(e) = source(l)
. . . | | |         ⇒
. . . | | |         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);[<tg,λs,v. f(s(x),v)>])))))
. . . | | | 16. D' : Dsys
. . . | | | 17. D ⊆ D'
. . . | | | 18. ∀w:World, p:FairFifo.
. . . | | |     PossibleWorld(D';w)
. . . | | |     ⇒
. . . | | |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | |     & (∀e:E.
. . . | | |        loc(e) = source(l)
. . . | | |        ⇒
. . . | | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                  kind(e))))
. . . | | |     & (∀e:E.
. . . | | |        isrcv(e)
. . . | | |        ⇒
. . . | | |        lnk(e) = l
. . . | | |        ⇒
```

```
. . . | | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                  kind(e))))
. . . | | |     & (∀e:E.
. . . | | |        loc(e) = source(l)
. . . | | |        ⇒
. . . | | |        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);[<tg,λs,v. f(s(x),v)>])))
. . . | | | 19. w : World
. . . | | | 20. ∀p:FairFifo.
. . . | | |     PossibleWorld(D';w)
. . . | | |     ⇒
. . . | | |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | |     & (∀e:E.
. . . | | |        loc(e) = source(l)
. . . | | |        ⇒
. . . | | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                  kind(e))))
. . . | | |     & (∀e:E.
. . . | | |        isrcv(e)
. . . | | |        ⇒
. . . | | |        lnk(e) = l
. . . | | |        ⇒
. . . | | |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                  kind(e))))
. . . | | |     & (∀e:E.
. . . | | |        loc(e) = source(l)
. . . | | |        ⇒
. . . | | |        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);[<tg,λs,v. f(s(x),v)>])))
. . . | | | 21. p : FairFifo
. . . | | | 22. PossibleWorld(D';w)
. . . | | | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | | | 24. ∀e:E.
. . . | | |     loc(e) = source(l)
. . . | | |     ⇒
```

```
. . . | | |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                kind(e)))
. . . | | | 25. ∀e:E.
. . . | | |     isrcv(e)
. . . | | |     ⇒
. . . | | |     lnk(e) = l
. . . | | |     ⇒
. . . | | |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | |                                kind(e)))
. . . | | | 26. ∀e:E.
. . . | | |     loc(e) = source(l)
. . . | | |     ⇒
. . . | | |     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);[<tg,λs,v. f(s(x),v)>]))
. . . | | | 27. vartype(source(l);x) ⊆r A
. . . | | | 28. ∀e:E.
. . . | | |     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | | | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | | | 30. e : E
. . . | | | 31. loc(e) = source(l)
. . . | | | 32. kind(e) = k
. . . | | | 33. L : {e':E| isrcv(e') & lnk(e') = l } List

. . . | | | 34. ∀e':E.
. . . | | |     (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | | | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | | | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | | |     =
. . . | | |     tagged-list-messages(λz.
. . . | | |     (z when e);val(e);[<tg,λs,v. f(s(x),v)>])
. . . | | | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | | | 38. e' : E
. . . | | | 39. (e' ∈ L) ⇒ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | | | 40. (e' ∈ L) ⇐ (isrcv(e') & lnk(e') = l & sender(e') = e)
. . . | | | 41. kind(e') = rcv(l; tg)
. . . | | | 42. sender(e') = e
. . . | | | ⊢  (e' ∈ L)
. . . | | |   by FwdThru
. . . | | |      Thm* ∀es:ES, l:IdLnk, tg:Id, e:E.
. . . | | |           kind(e) = rcv(l; tg)
. . . | | |           ⇒
. . . | | |           isrcv(e)
. . . | | |           & lnk(e) = l
. . . | | |             & tag(e) = tg
. . . | | |             & loc(sender(e)) = source(l)
. . . | | |      {es_DASH_kind_DASH_rcv}
```

```
. . . | | |      [-2]
. . . | | |      THEN
. . . | | |      ExRepD
. . . | | |      THEN
. . . | | |      BackThruSomeHyp
. . . | | |
. . . | | .\
. . . | | . 1. x : Id
. . . | | . 2. tg : Id
. . . | | . 3. k : Knd
. . . | | . 4. l : IdLnk
. . . | | . 5. T : Type
. . . | | . 6. A : Type
. . . | | . 7. B : Type
. . . | | . 8. f : A→B→(T List)
. . . | | . 9. rcv(l; tg) = k ⇒ T = B
. . . | | . 10. @source(l): ma-single-sends(x : A;
. . . | | .                                 k : B ⊕ rcv(l; tg) : T;
. . . | | .                                 k;
. . . | | .                                 l;
. . . | | .                                 [<tg,λs,v. f(s(x),v)>])
. . . | | .     ∈ Dsys
. . . | | . 11. ∀D:Dsys.
. . . | | .     @source(l): ma-single-sends(x : A;
. . . | | .                                 k : B ⊕ rcv(l; tg) : T;
. . . | | .                                 k;
. . . | | .                                 l;
. . . | | .                                 [<tg,λs,v. f(s(x),v)>]
. . . | | .                                 ) ⊆ D
. . . | | .     ⇒
. . . | | .     D
. . . | | .     realizes es.(∀x@0:Id.
. . . | | .                  vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | .                & (∀e:E.
. . . | | .                   loc(e) = source(l)
. . . | | .                   ⇒
. . . | | .                   (valtype(e) ⊆r ma-valtype
. . . | | .                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | | .                & (∀e:E.
. . . | | .                   isrcv(e)
. . . | | .                   ⇒
. . . | | .                   lnk(e) = l
. . . | | .                   ⇒
. . . | | .                   (valtype(e) ⊆r ma-valtype
. . . | | .                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | | .                & (∀e:E.
. . . | | .                   loc(e) = source(l)
. . . | | .                   ⇒
. . . | | .                   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);[<tg
. . . | | .                                           ,λs,v. f
. . . | | .                                                  (s(x)
. . . | | .                                                  ,v)>])))
. . . | | . 12. @source(l): ma-single-sends(x : A;
. . . | | .                                   k : B ⊕ rcv(l; tg) : T;
. . . | | .                                   k;
. . . | | .                                   l;
. . . | | .                                   [<tg,λs,v. f(s(x),v)>])
. . . | | .     ∈ Dsys
. . . | | . 13. D : Dsys
. . . | | . 14. @source(l): ma-single-sends(x : A;
. . . | | .                                   k : B ⊕ rcv(l; tg) : T;
. . . | | .                                   k;

. . . | | .                                   l;
. . . | | .                                   [<tg,λs,v. f(s(x),v)>]
. . . | | .                                   ) ⊆ D
. . . | | . 15. ∀D':Dsys.
. . . | | .     D ⊆ D'
. . . | | .     ⇒
. . . | | .     (∀w:World, p:FairFifo.
. . . | | .      PossibleWorld(D';w)
. . . | | .      ⇒
. . . | | .      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | .      & (∀e:E.
. . . | | .         loc(e) = source(l)
. . . | | .         ⇒
. . . | | .         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | .                                   kind(e))))
. . . | | .      & (∀e:E.
. . . | | .         isrcv(e)
. . . | | .         ⇒
. . . | | .         lnk(e) = l
. . . | | .         ⇒
. . . | | .         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | .                                   kind(e))))
. . . | | .      & (∀e:E.
. . . | | .         loc(e) = source(l)
. . . | | .         ⇒
. . . | | .         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);[<tg,λs,v. f(s(x),v)>])))
. . . | | . 16. D' : Dsys
. . . | | . 17. D ⊆ D'
. . . | | . 18. ∀w:World, p:FairFifo.
. . . | | .     PossibleWorld(D';w)
. . . | | .     ⇒
. . . | | .     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | .     & (∀e:E.
. . . | | .        loc(e) = source(l)
. . . | | .        ⇒
. . . | | .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | .                                  kind(e))))
. . . | | .     & (∀e:E.
. . . | | .        isrcv(e)
. . . | | .        ⇒
. . . | | .        lnk(e) = l
. . . | | .        ⇒
. . . | | .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | .                                  kind(e))))
. . . | | .     & (∀e:E.
. . . | | .        loc(e) = source(l)
. . . | | .        ⇒
. . . | | .        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);[<tg,λs,v. f(s(x),v)>])))
. . . | | . 19. w : World
. . . | | . 20. ∀p:FairFifo.
. . . | | .     PossibleWorld(D';w)
. . . | | .     ⇒
. . . | | .     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | | .     & (∀e:E.
. . . | | .        loc(e) = source(l)
. . . | | .        ⇒
. . . | | .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | .                                  kind(e))))
. . . | | .     & (∀e:E.
. . . | | .        isrcv(e)
. . . | | .        ⇒
. . . | | .        lnk(e) = l
. . . | | .        ⇒
. . . | | .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | .                                  kind(e))))
. . . | | .     & (∀e:E.
```

```
. . . | | .        loc(e) = source(l)
. . . | | .        ⇒
. . . | | .        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);[<tg,λs,v. f(s(x),v)>])))
. . . | | . 21. p : FairFifo
. . . | | . 22. PossibleWorld(D';w)
. . . | | . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | | . 24. ∀e:E.
. . . | | .     loc(e) = source(l)
. . . | | .     ⇒
. . . | | .     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | .                               kind(e)))
. . . | | . 25. ∀e:E.
. . . | | .     isrcv(e)
. . . | | .     ⇒
. . . | | .     lnk(e) = l
. . . | | .     ⇒
. . . | | .     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | | .                               kind(e)))
. . . | | . 26. ∀e:E.
. . . | | .     loc(e) = source(l)
. . . | | .     ⇒
. . . | | .     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);[<tg,λs,v. f(s(x),v)>]))
. . . | | . 27. vartype(source(l);x) ⊆r A
. . . | | . 28. ∀e:E.
. . . | | .     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | | . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | | . 30. e : E
. . . | | . 31. loc(e) = source(l)
. . . | | . 32. kind(e) = k
. . . | | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | | . 34. ∀e':E.
. . . | | .     (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
```

```
. . . | | . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | | .     =
. . . | | .     tagged-list-messages(λz.
. . . | | .     (z when e);val(e);[<tg,λs,v. f(s(x),v)>])
. . . | | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | | . 38. e' : E
. . . | | . 39. (e' ∈ L) ⇒ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | | . 40. (e' ∈ L) ⇐ (isrcv(e') & lnk(e') = l & sender(e') = e)
. . . | | . 41. kind(e') = rcv(l; tg)
. . . | | . ⊢  isrcv(e')
. . . | | .   by FwdThru
. . . | | .      Thm* ∀es:ES, l:IdLnk, tg:Id, e:E.
. . . | | .           kind(e) = rcv(l; tg)
. . . | | .           ⇒
. . . | | .           isrcv(e)
. . . | | .           & lnk(e) = l
. . . | | .             & tag(e) = tg
. . . | | .             & loc(sender(e)) = source(l)
. . . | | .      {es_DASH_kind_DASH_rcv}
. . . | | .      [-1]
. . . | | .      THEN
. . . | | .      ExRepD
. . . | |
. . . | .\
. . . | . 1. x : Id
. . . | . 2. tg : Id
. . . | . 3. k : Knd
. . . | . 4. l : IdLnk
. . . | . 5. T : Type
. . . | . 6. A : Type
. . . | . 7. B : Type
. . . | . 8. f : A→B→(T List)
. . . | . 9. rcv(l; tg) = k ⇒ T = B
. . . | . 10. @source(l): ma-single-sends(x : A;
. . . | . |                              k : B ⊕ rcv(l; tg) : T;
. . . | . |                              k;
. . . | . |                              l;
. . . | . |                              [<tg,λs,v. f(s(x),v)>])
. . . | . |   ∈ Dsys
. . . | . 11. ∀D:Dsys.
. . . | . |   @source(l): ma-single-sends(x : A;
. . . | . |                              k : B ⊕ rcv(l; tg) : T;
. . . | . |                              k;
. . . | . |                              l;
. . . | . |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . |   ⇒
. . . | . |   D
. . . | . |   realizes es.(∀x@0:Id.
. . . | . |                vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . |              & (∀e:E.
. . . | . |                 loc(e) = source(l)
. . . | . |                 ⇒
. . . | . |                 (valtype(e) ⊆r ma-valtype
. . . | . |                  (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . |              & (∀e:E.
```

```
. . . | . |                  isrcv(e)
. . . | . |                  ⇒

. . . | . |                  lnk(e) = l
. . . | . |                  ⇒
. . . | . |                  (valtype(e) ⊆r ma-valtype
. . . | . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . |               & (∀e:E.
. . . | . |                  loc(e) = source(l)
. . . | . |                  ⇒
. . . | . |                  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);[<tg
. . . | . |                                        ,λs,v. f
. . . | . |                                               (s(x)
. . . | . |                                               ,v)>])))
. . . | . 12. @source(l): ma-single-sends(x : A;
. . . | . |                                 k : B ⊕ rcv(l; tg) : T;
. . . | . |                                 k;
. . . | . |                                 l;
. . . | . |                                 [<tg,λs,v. f(s(x),v)>])
. . . | . |   ∈ Dsys
. . . | . 13. D : Dsys
. . . | . 14. @source(l): ma-single-sends(x : A;
. . . | . |                                 k : B ⊕ rcv(l; tg) : T;
. . . | . |                                 k;
. . . | . |                                 l;
. . . | . |                                 [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . 15. ∀D':Dsys.
. . . | . |   D ⊆ D'
. . . | . |   ⇒
. . . | . |   (∀w:World, p:FairFifo.
. . . | . |    PossibleWorld(D';w)
. . . | . |    ⇒
. . . | . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . |    & (∀e:E.
. . . | . |       loc(e) = source(l)
. . . | . |       ⇒
. . . | . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                  kind(e))))
. . . | . |    & (∀e:E.
. . . | . |       isrcv(e)
. . . | . |       ⇒
```

```
. . . | . |       lnk(e) = l
. . . | . |       ⇒
. . . | . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                  kind(e))))
. . . | . |    & (∀e:E.
. . . | . |       loc(e) = source(l)
. . . | . |       ⇒
. . . | . |       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);[<tg,λs,v. f(s(x),v)>])))))
. . . | . 16. D' : Dsys
. . . | . 17. D ⊆ D'
. . . | . 18. ∀w:World, p:FairFifo.
. . . | . |   PossibleWorld(D';w)
. . . | . |   ⇒
. . . | . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . |   & (∀e:E.
. . . | . |      loc(e) = source(l)
. . . | . |      ⇒
. . . | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                  kind(e))))
. . . | . |   & (∀e:E.
. . . | . |      isrcv(e)
. . . | . |      ⇒
. . . | . |      lnk(e) = l
. . . | . |      ⇒
. . . | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                  kind(e))))
. . . | . |   & (∀e:E.
. . . | . |      loc(e) = source(l)
. . . | . |      ⇒
. . . | . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . 19. w : World
. . . | . 20. ∀p:FairFifo.
. . . | . |   PossibleWorld(D';w)
```

```
. . . | . |   ⇒
. . . | . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . |   & (∀e:E.
. . . | . |      loc(e) = source(l)
. . . | . |      ⇒
. . . | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                kind(e))))
. . . | . |   & (∀e:E.
. . . | . |      isrcv(e)
. . . | . |      ⇒
. . . | . |      lnk(e) = l
. . . | . |      ⇒
. . . | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                kind(e))))
. . . | . |   & (∀e:E.
. . . | . |      loc(e) = source(l)
. . . | . |      ⇒
. . . | . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . 21. p : FairFifo
. . . | . 22. PossibleWorld(D';w)
. . . | . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . 24. ∀e:E.
. . . | . |   loc(e) = source(l)
. . . | . |   ⇒
. . . | . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                            kind(e)))
. . . | . 25. ∀e:E.
. . . | . |   isrcv(e)
. . . | . |   ⇒
. . . | . |   lnk(e) = l
. . . | . |   ⇒
. . . | . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                            kind(e)))
. . . | . 26. ∀e:E.
. . . | . |   loc(e) = source(l)
. . . | . |   ⇒
. . . | . |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | . 27. vartype(source(l);x) ⊆r A
. . . | . 28. ∀e:E.
. . . | . |   loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . 30. e : E
. . . | . 31. loc(e) = source(l)
. . . | . 32. kind(e) = k
. . . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . 34. ∀e':E.
. . . | . |   (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . |   =
. . . | . |   tagged-list-messages(λz.(z when e);val(e);[<tg
. . . | . |                                              ,λs,v. f
. . . | . |                                                     (s(x)
. . . | . |                                                     ,v)>])
. . . | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . ⊢  map(λe'.val(e');L) = f((x when e),val(e))
. . . | . | by Unfold `tagged-list-messages` -2
. . . | . |    THEN
. . . | . |    Unfold `concat` -2
. . . | . |    THEN
. . . | . |    Reduce -2
. . . | . |
. . . | . 1. x : Id
. . . | . 2. tg : Id
. . . | . 3. k : Knd
. . . | . 4. l : IdLnk
. . . | . 5. T : Type
. . . | . 6. A : Type
. . . | . 7. B : Type
. . . | . 8. f : A→B→(T List)
. . . | . 9. rcv(l; tg) = k ⇒ T = B
. . . | . 10. @source(l): ma-single-sends(x : A;
. . . | . |                              k : B ⊕ rcv(l; tg) : T;
. . . | . |                              k;
. . . | . |                              l;
. . . | . |                              [<tg,λs,v. f(s(x),v)>])
. . . | . |   ∈ Dsys
. . . | . 11. ∀D:Dsys.
. . . | . |   @source(l): ma-single-sends(x : A;
. . . | . |                              k : B ⊕ rcv(l; tg) : T;
. . . | . |                              k;
. . . | . |                              l;
. . . | . |                              [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . |   ⇒
. . . | . |   D
. . . | . |   realizes es.(∀x@0:Id.
. . . | . |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . |               & (∀e:E.
. . . | . |                  loc(e) = source(l)
. . . | . |                  ⇒
```

```
. . . | . |                  (valtype(e) ⊆r ma-valtype
. . . | . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . |               & (∀e:E.
. . . | . |                  isrcv(e)
. . . | . |                  ⇒
. . . | . |                  lnk(e) = l
. . . | . |                  ⇒
. . . | . |                  (valtype(e) ⊆r ma-valtype
. . . | . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . |               & (∀e:E.
. . . | . |                  loc(e) = source(l)
. . . | . |                  ⇒
. . . | . |                  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);[<tg
. . . | . |                                       ,λs,v. f
. . . | . |                                              (s(x)
. . . | . |                                              ,v)>])))
. . . | . 12. @source(l): ma-single-sends(x : A;
. . . | . |                                 k : B ⊕ rcv(l; tg) : T;
. . . | . |                                 k;
. . . | . |                                 l;
. . . | . |                                 [<tg,λs,v. f(s(x),v)>])
. . . | . |   ∈ Dsys
. . . | . 13. D : Dsys
. . . | . 14. @source(l): ma-single-sends(x : A;
. . . | . |                                 k : B ⊕ rcv(l; tg) : T;
. . . | . |                                 k;
. . . | . |                                 l;
. . . | . |                                 [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . 15. ∀D':Dsys.
. . . | . |   D ⊆ D'
. . . | . |   ⇒
. . . | . |   (∀w:World, p:FairFifo.
. . . | . |    PossibleWorld(D';w)
. . . | . |    ⇒
. . . | . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . |    & (∀e:E.
. . . | . |       loc(e) = source(l)
. . . | . |       ⇒
. . . | . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                 kind(e))))
```

```
. . . | . |    & (∀e:E.
. . . | . |       isrcv(e)
. . . | . |       ⇒
. . . | . |       lnk(e) = l
. . . | . |       ⇒
. . . | . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                  kind(e))))
. . . | . |    & (∀e:E.
. . . | . |       loc(e) = source(l)
. . . | . |       ⇒
. . . | . |       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);[<tg,λs,v. f(s(x),v)>])))))
. . . | . 16. D' : Dsys
. . . | . 17. D ⊆ D'
. . . | . 18. ∀w:World, p:FairFifo.
. . . | . |   PossibleWorld(D';w)
. . . | . |   ⇒
. . . | . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . |   & (∀e:E.
. . . | . |      loc(e) = source(l)
. . . | . |      ⇒
. . . | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                 kind(e))))
. . . | . |   & (∀e:E.
. . . | . |      isrcv(e)
. . . | . |      ⇒
. . . | . |      lnk(e) = l
. . . | . |      ⇒
. . . | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                 kind(e))))
. . . | . |   & (∀e:E.
. . . | . |      loc(e) = source(l)
. . . | . |      ⇒
. . . | . |      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);[<tg,λs,v. f(s(x),v)>])))
```

```
. . . | . 19. w : World
. . . | . 20. ∀p:FairFifo.
. . . | . |   PossibleWorld(D';w)
. . . | . |   ⇒
. . . | . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . |   & (∀e:E.
. . . | . |      loc(e) = source(l)
. . . | . |      ⇒
. . . | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                kind(e))))
. . . | . |   & (∀e:E.
. . . | . |      isrcv(e)
. . . | . |      ⇒
. . . | . |      lnk(e) = l
. . . | . |      ⇒
. . . | . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                kind(e))))
. . . | . |   & (∀e:E.
. . . | . |      loc(e) = source(l)
. . . | . |      ⇒
. . . | . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . 21. p : FairFifo
. . . | . 22. PossibleWorld(D';w)
. . . | . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . 24. ∀e:E.
. . . | . |   loc(e) = source(l)
. . . | . |   ⇒
. . . | . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                            kind(e)))
. . . | . 25. ∀e:E.
. . . | . |   isrcv(e)
. . . | . |   ⇒
. . . | . |   lnk(e) = l
. . . | . |   ⇒
. . . | . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                            kind(e)))
. . . | . 26. ∀e:E.
. . . | . |   loc(e) = source(l)
. . . | . |   ⇒
. . . | . |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | . 27. vartype(source(l);x) ⊆r A
. . . | . 28. ∀e:E.
. . . | . |   loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . 30. e : E
. . . | . 31. loc(e) = source(l)
. . . | . 32. kind(e) = k
. . . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . 34. ∀e':E.
. . . | . |   (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . |   =
. . . | . |   (map(λx@0.<tg,x@0>;f((x when e),val(e))) @ nil)
. . . | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . ⊢  map(λe'.val(e');L) = f((x when e),val(e))
. . . | .  | by Assert (f((x when e),val(e)) ∈ T List)
. . . | .  |
. . . | .  |\
. . . | .  | .....assertion .....
. . . | .  | 1. x : Id
. . . | .  | 2. tg : Id
. . . | .  | 3. k : Knd
. . . | .  | 4. l : IdLnk
. . . | .  | 5. T : Type
. . . | .  | 6. A : Type
. . . | .  | 7. B : Type
. . . | .  | 8. f : A→B→(T List)
. . . | .  | 9. rcv(l; tg) = k ⇒ T = B
. . . | .  | 10. @source(l): ma-single-sends(x : A;
. . . | .  |                                k : B ⊕ rcv(l; tg) : T;
. . . | .  |                                k;
. . . | .  |                                l;
. . . | .  |                                [<tg,λs,v. f(s(x),v)>])
. . . | .  |     ∈ Dsys
. . . | .  | 11. ∀D:Dsys.
. . . | .  |     @source(l): ma-single-sends(x : A;
. . . | .  |                                k : B ⊕ rcv(l; tg) : T;
. . . | .  |                                k;
. . . | .  |                                l;
. . . | .  |                                [<tg,λs,v. f(s(x),v)>]
. . . | .  |                                ) ⊆ D
. . . | .  |     ⇒
. . . | .  |     D
. . . | .  |     realizes es.(∀x@0:Id.
. . . | .  |                  vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | .  |                 & (∀e:E.
. . . | .  |                    loc(e) = source(l)
. . . | .  |                    ⇒
. . . | .  |                    (valtype(e) ⊆r ma-valtype
```

```
. . . | . |                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . |                & (∀e:E.
. . . | . |                   isrcv(e)
. . . | . |                   ⇒
. . . | . |                   lnk(e) = l
. . . | . |                   ⇒
. . . | . |                   (valtype(e) ⊆r ma-valtype
. . . | . |                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . |                & (∀e:E.
. . . | . |                   loc(e) = source(l)
. . . | . |                   ⇒
. . . | . |                   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);[<tg
. . . | . |                                        ,λs,v. f
. . . | . |                                               (s(x)
. . . | . |                                               ,v)>])))
. . . | . | 12. @source(l): ma-single-sends(x : A;
. . . | . |                                  k : B ⊕ rcv(l; tg) : T;
. . . | . |                                  k;
. . . | . |                                  l;
. . . | . |                                  [<tg,λs,v. f(s(x),v)>])
. . . | . |     ∈ Dsys
. . . | . | 13. D : Dsys
. . . | . | 14. @source(l): ma-single-sends(x : A;
. . . | . |                                  k : B ⊕ rcv(l; tg) : T;
. . . | . |                                  k;
. . . | . |                                  l;
. . . | . |                                  [<tg,λs,v. f(s(x),v)>]
. . . | . |                                  ) ⊆ D
. . . | . | 15. ∀D':Dsys.
. . . | . |     D ⊆ D'
. . . | . |     ⇒
. . . | . |     (∀w:World, p:FairFifo.
. . . | . |      PossibleWorld(D';w)
. . . | . |      ⇒
. . . | . |      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . |      & (∀e:E.
. . . | . |         loc(e) = source(l)
. . . | . |         ⇒
. . . | . |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                   kind(e))))
```

```
. . . | . |      & (∀e:E.
. . . | . |         isrcv(e)
. . . | . |         ⇒
. . . | . |         lnk(e) = l
. . . | . |         ⇒
. . . | . |         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                   kind(e))))
. . . | . |      & (∀e:E.
. . . | . |         loc(e) = source(l)
. . . | . |         ⇒
. . . | . |         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);[<tg,λs,v. f(s(x),v)>])))))
. . . | . | 16. D' : Dsys
. . . | . | 17. D ⊆ D'
. . . | . | 18. ∀w:World, p:FairFifo.
. . . | . |     PossibleWorld(D';w)
. . . | . |     ⇒
. . . | . |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . |     & (∀e:E.
. . . | . |        loc(e) = source(l)
. . . | . |        ⇒
. . . | . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                   kind(e))))
. . . | . |     & (∀e:E.
. . . | . |        isrcv(e)
. . . | . |        ⇒
. . . | . |        lnk(e) = l
. . . | . |        ⇒
. . . | . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                   kind(e))))
. . . | . |     & (∀e:E.
. . . | . |        loc(e) = source(l)
. . . | . |        ⇒
. . . | . |        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);[<tg,λs,v. f(s(x),v)>])))
```

```
. . . | . | 19. w : World
. . . | . | 20. ∀p:FairFifo.
. . . | . |     PossibleWorld(D';w)
. . . | . |     ⇒
. . . | . |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . |     & (∀e:E.
. . . | . |        loc(e) = source(l)
. . . | . |        ⇒
. . . | . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                  kind(e))))
. . . | . |     & (∀e:E.
. . . | . |        isrcv(e)
. . . | . |        ⇒
. . . | . |        lnk(e) = l
. . . | . |        ⇒
. . . | . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                  kind(e))))
. . . | . |     & (∀e:E.
. . . | . |        loc(e) = source(l)
. . . | . |        ⇒
. . . | . |        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);[<tg,λs,v. f(s(x),v)>])))
. . . | . | 21. p : FairFifo
. . . | . | 22. PossibleWorld(D';w)
. . . | . | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . | 24. ∀e:E.
. . . | . |     loc(e) = source(l)
. . . | . |     ⇒
. . . | . |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                kind(e)))
. . . | . | 25. ∀e:E.
. . . | . |     isrcv(e)
. . . | . |     ⇒
. . . | . |     lnk(e) = l
. . . | . |     ⇒
. . . | . |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . |                                kind(e)))
. . . | . | 26. ∀e:E.
. . . | . |     loc(e) = source(l)
. . . | . |     ⇒
. . . | . |     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);[<tg,λs,v. f(s(x),v)>]))
. . . | . | 27. vartype(source(l);x) ⊆r A
. . . | . | 28. ∀e:E.
. . . | . |     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . | 30. e : E
. . . | . | 31. loc(e) = source(l)
. . . | . | 32. kind(e) = k
. . . | . | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . | 34. ∀e':E.
. . . | . |     (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . |     =
. . . | . |     (map(λx@0.<tg,x@0>;f((x when e),val(e))) @ nil)
. . . | . | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . | ⊢  f((x when e),val(e)) ∈ T List  by MaAuto
. . . | . |
. . . | . .\
. . . | . . 1. x : Id
. . . | . . 2. tg : Id
. . . | . . 3. k : Knd
. . . | . . 4. l : IdLnk
. . . | . . 5. T : Type
. . . | . . 6. A : Type
. . . | . . 7. B : Type
. . . | . . 8. f : A→B→(T List)
. . . | . . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . 10. @source(l): ma-single-sends(x : A;
. . . | . . |                              k : B ⊕ rcv(l; tg) : T;
. . . | . . |                              k;
. . . | . . |                              l;
. . . | . . |                              [<tg,λs,v. f(s(x),v)>])
. . . | . . |   ∈ Dsys
. . . | . . 11. ∀D:Dsys.
. . . | . . |   @source(l): ma-single-sends(x : A;
. . . | . . |                              k : B ⊕ rcv(l; tg) : T;
. . . | . . |                              k;
. . . | . . |                              l;
. . . | . . |                              [<tg,λs,v. f(s(x),v)>]
. . . | . . |                              ) ⊆ D
. . . | . . |   ⇒
. . . | . . |   D
. . . | . . |   realizes es.(∀x@0:Id.
. . . | . . |                vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . |               & (∀e:E.
. . . | . . |                  loc(e) = source(l)
. . . | . . |                  ⇒
. . . | . . |                  (valtype(e) ⊆r ma-valtype
```

```
. . . | . . |                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . |                & (∀e:E.
. . . | . . |                   isrcv(e)
. . . | . . |                   ⇒
. . . | . . |                   lnk(e) = l
. . . | . . |                   ⇒
. . . | . . |                   (valtype(e) ⊆r ma-valtype
. . . | . . |                    (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . |                & (∀e:E.
. . . | . . |                   loc(e) = source(l)
. . . | . . |                   ⇒
. . . | . . |                   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);[<tg
. . . | . . |                                         ,λs,v. f
. . . | . . |                                                (s(x)
. . . | . . |                                                ,v)>])))
. . . | . . 12. @source(l): ma-single-sends(x : A;
. . . | . . |                                k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                k;
. . . | . . |                                l;
. . . | . . |                                [<tg,λs,v. f(s(x),v)>])
. . . | . . |   ∈ Dsys
. . . | . . 13. D : Dsys
. . . | . . 14. @source(l): ma-single-sends(x : A;
. . . | . . |                                k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                k;
. . . | . . |                                l;
. . . | . . |                                [<tg,λs,v. f(s(x),v)>]
. . . | . . |                                ) ⊆ D
. . . | . . 15. ∀D':Dsys.
. . . | . . |   D ⊆ D'
. . . | . . |   ⇒
. . . | . . |   (∀w:World, p:FairFifo.
. . . | . . |    PossibleWorld(D';w)
. . . | . . |    ⇒
. . . | . . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . |    & (∀e:E.
. . . | . . |       loc(e) = source(l)
. . . | . . |       ⇒
. . . | . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                 kind(e))))
```

```
. . . | . . |    & (∀e:E.
. . . | . . |       isrcv(e)
. . . | . . |       ⇒
. . . | . . |       lnk(e) = l
. . . | . . |       ⇒
. . . | . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                 kind(e))))
. . . | . . |    & (∀e:E.
. . . | . . |       loc(e) = source(l)
. . . | . . |       ⇒
. . . | . . |       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);[<tg,λs,v. f(s(x),v)>])))))
. . . | . . 16. D' : Dsys
. . . | . . 17. D ⊆ D'
. . . | . . 18. ∀w:World, p:FairFifo.
. . . | . . |   PossibleWorld(D';w)
. . . | . . |   ⇒
. . . | . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . |   & (∀e:E.
. . . | . . |      loc(e) = source(l)
. . . | . . |      ⇒
. . . | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                 kind(e))))
. . . | . . |   & (∀e:E.
. . . | . . |      isrcv(e)
. . . | . . |      ⇒
. . . | . . |      lnk(e) = l
. . . | . . |      ⇒
. . . | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                 kind(e))))
. . . | . . |   & (∀e:E.
. . . | . . |      loc(e) = source(l)
. . . | . . |      ⇒
. . . | . . |      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);[<tg,λs,v. f(s(x),v)>])))
```

```
. . . | . . 19. w : World
. . . | . . 20. ∀p:FairFifo.
. . . | . . |   PossibleWorld(D';w)
. . . | . . |   ⇒
. . . | . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . |   & (∀e:E.
. . . | . . |      loc(e) = source(l)
. . . | . . |      ⇒
. . . | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                kind(e))))
. . . | . . |   & (∀e:E.
. . . | . . |      isrcv(e)
. . . | . . |      ⇒
. . . | . . |      lnk(e) = l
. . . | . . |      ⇒
. . . | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                kind(e))))
. . . | . . |   & (∀e:E.
. . . | . . |      loc(e) = source(l)
. . . | . . |      ⇒
. . . | . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . 21. p : FairFifo
. . . | . . 22. PossibleWorld(D';w)
. . . | . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . 24. ∀e:E.
. . . | . . |   loc(e) = source(l)
. . . | . . |   ⇒
. . . | . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                            kind(e)))
. . . | . . 25. ∀e:E.
. . . | . . |   isrcv(e)
. . . | . . |   ⇒
. . . | . . |   lnk(e) = l
. . . | . . |   ⇒
. . . | . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                            kind(e)))
. . . | . . 26. ∀e:E.
. . . | . . |   loc(e) = source(l)
. . . | . . |   ⇒
. . . | . . |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . 27. vartype(source(l);x) ⊆r A
. . . | . . 28. ∀e:E.
. . . | . . |   loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . 30. e : E
. . . | . . 31. loc(e) = source(l)
. . . | . . 32. kind(e) = k
. . . | . . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . 34. ∀e':E.
. . . | . . |   (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . |   =
. . . | . . |   (map(λx@0.<tg,x@0>;f((x when e),val(e))) @ nil)
. . . | . . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . 38. f((x when e),val(e)) ∈ T List
. . . | . . ⊢  map(λe'.val(e');L) = f((x when e),val(e))
. . . | . . | by RW0 Thm* ∀l:T List. (l @ nil) ∼ l    {append_nil_sq}
. . . | . . |    -3
. . . | . . |
. . . | . . 1. x : Id
. . . | . . 2. tg : Id
. . . | . . 3. k : Knd
. . . | . . 4. l : IdLnk
. . . | . . 5. T : Type
. . . | . . 6. A : Type
. . . | . . 7. B : Type
. . . | . . 8. f : A→B→(T List)
. . . | . . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . 10. @source(l): ma-single-sends(x : A;
. . . | . . |                              k : B ⊕ rcv(l; tg) : T;
. . . | . . |                              k;
. . . | . . |                              l;
. . . | . . |                              [<tg,λs,v. f(s(x),v)>])
. . . | . . |   ∈ Dsys
. . . | . . 11. ∀D:Dsys.
. . . | . . |   @source(l): ma-single-sends(x : A;
. . . | . . |                              k : B ⊕ rcv(l; tg) : T;
. . . | . . |                              k;
. . . | . . |                              l;
. . . | . . |                              [<tg,λs,v. f(s(x),v)>]
. . . | . . |                              ) ⊆ D
. . . | . . |   ⇒
. . . | . . |   D
. . . | . . |   realizes es.(∀x@0:Id.
. . . | . . |                vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . |               & (∀e:E.
. . . | . . |                  loc(e) = source(l)
```

```
. . . | . . |                  ⇒
. . . | . . |                  (valtype(e) ⊆r ma-valtype
. . . | . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . |               & (∀e:E.
. . . | . . |                  isrcv(e)
. . . | . . |                  ⇒
. . . | . . |                  lnk(e) = l
. . . | . . |                  ⇒
. . . | . . |                  (valtype(e) ⊆r ma-valtype
. . . | . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . |               & (∀e:E.
. . . | . . |                  loc(e) = source(l)
. . . | . . |                  ⇒
. . . | . . |                  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);[<tg
. . . | . . |                                        ,λs,v. f
. . . | . . |                                               (s(x)
. . . | . . |                                               ,v)>])))
. . . | . . 12. @source(l): ma-single-sends(x : A;
. . . | . . |                                k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                k;
. . . | . . |                                l;
. . . | . . |                                [<tg,λs,v. f(s(x),v)>])
. . . | . . |   ∈ Dsys
. . . | . . 13. D : Dsys
. . . | . . 14. @source(l): ma-single-sends(x : A;
. . . | . . |                                k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                k;
. . . | . . |                                l;
. . . | . . |                                [<tg,λs,v. f(s(x),v)>]
. . . | . . |                                ) ⊆ D
. . . | . . 15. ∀D':Dsys.
. . . | . . |   D ⊆ D'
. . . | . . |   ⇒
. . . | . . |   (∀w:World, p:FairFifo.
. . . | . . |    PossibleWorld(D';w)
. . . | . . |    ⇒
. . . | . . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . |    & (∀e:E.
. . . | . . |       loc(e) = source(l)
. . . | . . |       ⇒
```

```
. . . | . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                  kind(e))))
. . . | . . |    & (∀e:E.
. . . | . . |       isrcv(e)
. . . | . . |       ⇒
. . . | . . |       lnk(e) = l
. . . | . . |       ⇒
. . . | . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                  kind(e))))
. . . | . . |    & (∀e:E.
. . . | . . |       loc(e) = source(l)
. . . | . . |       ⇒
. . . | . . |       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);[<tg,λs,v. f(s(x),v)>])))))
. . . | . . 16. D' : Dsys
. . . | . . 17. D ⊆ D'
. . . | . . 18. ∀w:World, p:FairFifo.
. . . | . . |   PossibleWorld(D';w)
. . . | . . |   ⇒
. . . | . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . |   & (∀e:E.
. . . | . . |      loc(e) = source(l)
. . . | . . |      ⇒
. . . | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                  kind(e))))
. . . | . . |   & (∀e:E.
. . . | . . |      isrcv(e)
. . . | . . |      ⇒
. . . | . . |      lnk(e) = l
. . . | . . |      ⇒
. . . | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                  kind(e))))
. . . | . . |   & (∀e:E.
. . . | . . |      loc(e) = source(l)
. . . | . . |      ⇒
. . . | . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . 19. w : World
. . . | . . 20. ∀p:FairFifo.
. . . | . . |   PossibleWorld(D';w)
. . . | . . |   ⇒
. . . | . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . |   & (∀e:E.
. . . | . . |      loc(e) = source(l)
. . . | . . |      ⇒
. . . | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                kind(e))))
. . . | . . |   & (∀e:E.
. . . | . . |      isrcv(e)
. . . | . . |      ⇒
. . . | . . |      lnk(e) = l
. . . | . . |      ⇒
. . . | . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                kind(e))))
. . . | . . |   & (∀e:E.
. . . | . . |      loc(e) = source(l)
. . . | . . |      ⇒
. . . | . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . 21. p : FairFifo
. . . | . . 22. PossibleWorld(D';w)
. . . | . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . 24. ∀e:E.
. . . | . . |   loc(e) = source(l)
. . . | . . |   ⇒
. . . | . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                            kind(e)))
. . . | . . 25. ∀e:E.
. . . | . . |   isrcv(e)
. . . | . . |   ⇒
. . . | . . |   lnk(e) = l
. . . | . . |   ⇒
. . . | . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                            kind(e)))
. . . | . . 26. ∀e:E.
. . . | . . |   loc(e) = source(l)
. . . | . . |   ⇒
. . . | . . |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . 27. vartype(source(l);x) ⊆r A
. . . | . . 28. ∀e:E.
. . . | . . |   loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . 30. e : E
. . . | . . 31. loc(e) = source(l)
. . . | . . 32. kind(e) = k
. . . | . . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . 34. ∀e':E.
. . . | . . |   (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . |   =
. . . | . . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . 38. f((x when e),val(e)) ∈ T List
. . . | . . ⊢  map(λe'.val(e');L) = f((x when e),val(e))
. . . | . . | by Subst
. . . | . . |    ((λe'.val(e')) ∼ ((λp.2of(p)) o (λe'.<tag(e')
. . . | . . |                                        ,val(e')>)))
. . . | . . |    0
. . . | . . |
. . . | . . |\
. . . | . . | .....equality .....
. . . | . . | 1. x : Id
. . . | . . | 2. tg : Id
. . . | . . | 3. k : Knd
. . . | . . | 4. l : IdLnk
. . . | . . | 5. T : Type
. . . | . . | 6. A : Type
. . . | . . | 7. B : Type
. . . | . . | 8. f : A→B→(T List)
. . . | . . | 9. rcv(l; tg) = k ⇒ T = B
. . . | . . | 10. @source(l): ma-single-sends(x : A;
. . . | . . |                                k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                k;
. . . | . . |                                l;
. . . | . . |                                [<tg,λs,v. f(s(x),v)>])
. . . | . . |     ∈ Dsys
. . . | . . | 11. ∀D:Dsys.
. . . | . . |     @source(l): ma-single-sends(x : A;
. . . | . . |                                k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                k;
. . . | . . |                                l;
. . . | . . |                                [<tg,λs,v. f(s(x),v)>]
. . . | . . |                                ) ⊆ D
```

```
. . . | . . |     ⇒
. . . | . . |     D

. . . | . . |     realizes es.(∀x@0:Id.
. . . | . . |                  vartype(source(l);x@0) ⊆r ...(x@0)?Top)
. . . | . . |                 & (∀e:E.
. . . | . . |                    loc(e) = source(l)
. . . | . . |                    ⇒
. . . | . . |                    (valtype(e) ⊆r ma-valtype
. . . | . . |                     (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . |                 & (∀e:E.
. . . | . . |                    isrcv(e)
. . . | . . |                    ⇒
. . . | . . |                    lnk(e) = l
. . . | . . |                    ⇒
. . . | . . |                    (valtype(e) ⊆r ma-valtype
. . . | . . |                     (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . |                 & (∀e:E.
. . . | . . |                    loc(e) = source(l)
. . . | . . |                    ⇒
. . . | . . |                    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);[<tg
. . . | . . |                                          ,λs,v. f
. . . | . . |                                                 (s(x)
. . . | . . |                                                 ,v)>])))
. . . | . . | 12. @source(l): ma-single-sends(x : A;
. . . | . . |                                 k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                 k;
. . . | . . |                                 l;
. . . | . . |                                 [<tg,λs,v. f(s(x),v)>])
. . . | . . |     ∈ Dsys
. . . | . . | 13. D : Dsys
. . . | . . | 14. @source(l): ma-single-sends(x : A;
. . . | . . |                                 k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                 k;
. . . | . . |                                 l;
. . . | . . |                                 [<tg,λs,v. f(s(x),v)>]
. . . | . . |                                 ) ⊆ D
. . . | . . | 15. ∀D':Dsys.
. . . | . . |     D ⊆ D'
. . . | . . |     ⇒
```

```
. . . | . . |     (∀w:World, p:FairFifo.
. . . | . . |      PossibleWorld(D';w)
. . . | . . |      ⇒
. . . | . . |      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . |      & (∀e:E.
. . . | . . |         loc(e) = source(l)
. . . | . . |         ⇒
. . . | . . |         (valtype(e) ⊆r ma-valtype
. . . | . . |          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . |      & (∀e:E.
. . . | . . |         isrcv(e)
. . . | . . |         ⇒
. . . | . . |         lnk(e) = l
. . . | . . |         ⇒
. . . | . . |         (valtype(e) ⊆r ma-valtype
. . . | . . |          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . |      & (∀e:E.
. . . | . . |         loc(e) = source(l)
. . . | . . |         ⇒
. . . | . . |         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);[<tg,λs,v. f(s(x),v)>])))))
. . . | . . | 16. D' : Dsys
. . . | . . | 17. D ⊆ D'
. . . | . . | 18. ∀w:World, p:FairFifo.
. . . | . . |     PossibleWorld(D';w)
. . . | . . |     ⇒
. . . | . . |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . |     & (∀e:E.
. . . | . . |        loc(e) = source(l)
. . . | . . |        ⇒
. . . | . . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                  kind(e))))
. . . | . . |     & (∀e:E.
. . . | . . |        isrcv(e)
. . . | . . |        ⇒
. . . | . . |        lnk(e) = l
. . . | . . |        ⇒

. . . | . . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                  kind(e))))
. . . | . . |     & (∀e:E.
. . . | . . |        loc(e) = source(l)
. . . | . . |        ⇒
. . . | . . |        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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . | 19. w : World
. . . | . . | 20. ∀p:FairFifo.
. . . | . . |     PossibleWorld(D';w)
. . . | . . |     ⇒
. . . | . . |     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . |     & (∀e:E.
. . . | . . |        loc(e) = source(l)
. . . | . . |        ⇒
. . . | . . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                  kind(e))))
. . . | . . |     & (∀e:E.
. . . | . . |        isrcv(e)
. . . | . . |        ⇒
. . . | . . |        lnk(e) = l
. . . | . . |        ⇒
. . . | . . |        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                  kind(e))))
. . . | . . |     & (∀e:E.
. . . | . . |        loc(e) = source(l)
. . . | . . |        ⇒
. . . | . . |        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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . | 21. p : FairFifo
. . . | . . | 22. PossibleWorld(D';w)
. . . | . . | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . | 24. ∀e:E.
. . . | . . |     loc(e) = source(l)
. . . | . . |     ⇒
. . . | . . |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . |                                kind(e)))
. . . | . . | 25. ∀e:E.
. . . | . . |     isrcv(e)
. . . | . . |     ⇒
. . . | . . |     lnk(e) = l
. . . | . . |     ⇒
. . . | . . |     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
```

```
. . . | . . |                                kind(e)))
. . . | . . | 26. ∀e:E.
. . . | . . |     loc(e) = source(l)
. . . | . . |     ⇒
. . . | . . |     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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . | 27. vartype(source(l);x) ⊆r A
. . . | . . | 28. ∀e:E.
. . . | . . |     loc(e) = source(l)
. . . | . . |     ⇒
. . . | . . |     kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . | 30. e : E
. . . | . . | 31. loc(e) = source(l)
. . . | . . | 32. kind(e) = k
. . . | . . | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . | 34. ∀e':E.
. . . | . . |     (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . |     =
. . . | . . |     map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . | 38. f((x when e),val(e)) ∈ T List
. . . | . . | ⊢  (λe'.val(e')) ∼ ((λp.2of(p))
. . . | . . |     o (λe'.<tag(e'),val(e')>))
. . . | . . |   by Unfold `compose` 0 THEN Reduce 0 THEN Trivial

. . . | . . |
. . . | . . .\
. . . | . . . 1. x : Id
. . . | . . . 2. tg : Id
. . . | . . . 3. k : Knd
. . . | . . . 4. l : IdLnk
. . . | . . . 5. T : Type
. . . | . . . 6. A : Type
. . . | . . . 7. B : Type
. . . | . . . 8. f : A→B→(T List)
. . . | . . . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . 10. @source(l): ma-single-sends(x : A;
. . . | . . . |                              k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                              k;
. . . | . . . |                              l;
. . . | . . . |                              [<tg,λs,v. f(s(x),v)>])
. . . | . . . |   ∈ Dsys
. . . | . . . 11. ∀D:Dsys.
```

```
. . . | . . . |   @source(l): ma-single-sends(x : A;
. . . | . . . |                               k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                               k;
. . . | . . . |                               l;
. . . | . . . |                               [<tg,λs,v. f(s(x),v)>]
. . . | . . . |                               ) ⊆ D
. . . | . . . |   ⇒
. . . | . . . |   D
. . . | . . . |   realizes es.(∀x@0:Id.
. . . | . . . |                vartype(source(l);x@0) ⊆r ...(x@0)?Top)
. . . | . . . |               & (∀e:E.
. . . | . . . |                  loc(e) = source(l)
. . . | . . . |                  ⇒
. . . | . . . |                  (valtype(e) ⊆r ma-valtype
. . . | . . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . |               & (∀e:E.
. . . | . . . |                  isrcv(e)
. . . | . . . |                  ⇒
. . . | . . . |                  lnk(e) = l
. . . | . . . |                  ⇒
. . . | . . . |                  (valtype(e) ⊆r ma-valtype
. . . | . . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . |               & (∀e:E.
. . . | . . . |                  loc(e) = source(l)
. . . | . . . |                  ⇒
. . . | . . . |                  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);[<tg
. . . | . . . |                                        ,λs,v. f
. . . | . . . |                                               (s(x)
. . . | . . . |                                               ,v)>])))
. . . | . . 12. @source(l): ma-single-sends(x : A;
. . . | . . . |                               k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                               k;
. . . | . . . |                               l;
. . . | . . . |                               [<tg,λs,v. f(s(x),v)>])
. . . | . . . |   ∈ Dsys
. . . | . . 13. D : Dsys
. . . | . . 14. @source(l): ma-single-sends(x : A;
. . . | . . . |                               k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                               k;
```

```
. . . | . . . |                                l;
. . . | . . . |                                [<tg,λs,v. f(s(x),v)>]
. . . | . . . |                                ) ⊆ D
. . . | . . . 15. ∀D':Dsys.
. . . | . . . |   D ⊆ D'
. . . | . . . |   ⇒
. . . | . . . |   (∀w:World, p:FairFifo.
. . . | . . . |    PossibleWorld(D';w)
. . . | . . . |    ⇒
. . . | . . . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . |    & (∀e:E.
. . . | . . . |       loc(e) = source(l)
. . . | . . . |       ⇒
. . . | . . . |       (valtype(e) ⊆r ma-valtype
. . . | . . . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . |    & (∀e:E.
. . . | . . . |       isrcv(e)
. . . | . . . |       ⇒
. . . | . . . |       lnk(e) = l
. . . | . . . |       ⇒
. . . | . . . |       (valtype(e) ⊆r ma-valtype
. . . | . . . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . |    & (∀e:E.
. . . | . . . |       loc(e) = source(l)
. . . | . . . |       ⇒
. . . | . . . |       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);[<tg,λs,v. f(s(x),v)>])))))
. . . | . . . 16. D' : Dsys
. . . | . . . 17. D ⊆ D'
. . . | . . . 18. ∀w:World, p:FairFifo.
. . . | . . . |   PossibleWorld(D';w)
. . . | . . . |   ⇒
. . . | . . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . |   & (∀e:E.
. . . | . . . |      loc(e) = source(l)
. . . | . . . |      ⇒
. . . | . . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                                kind(e))))
. . . | . . . |   & (∀e:E.
. . . | . . . |      isrcv(e)
. . . | . . . |      ⇒
. . . | . . . |      lnk(e) = l
. . . | . . . |      ⇒
. . . | . . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                                kind(e))))
```

```
. . . | . . . |   & (∀e:E.
. . . | . . . |      loc(e) = source(l)
. . . | . . . |      ⇒
. . . | . . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . . 19. w : World
. . . | . . . 20. ∀p:FairFifo.
. . . | . . . |   PossibleWorld(D';w)
. . . | . . . |   ⇒
. . . | . . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . |   & (∀e:E.
. . . | . . . |      loc(e) = source(l)
. . . | . . . |      ⇒
. . . | . . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                                kind(e))))
. . . | . . . |   & (∀e:E.
. . . | . . . |      isrcv(e)
. . . | . . . |      ⇒
. . . | . . . |      lnk(e) = l
. . . | . . . |      ⇒
. . . | . . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                                kind(e))))
. . . | . . . |   & (∀e:E.
. . . | . . . |      loc(e) = source(l)
. . . | . . . |      ⇒
. . . | . . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . . 21. p : FairFifo
. . . | . . . 22. PossibleWorld(D';w)
. . . | . . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . 24. ∀e:E.
. . . | . . . |   loc(e) = source(l)
. . . | . . . |   ⇒
. . . | . . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                            kind(e)))
```

```
. . . | . . . 25. ∀e:E.
. . . | . . . |   isrcv(e)
. . . | . . . |   ⇒
. . . | . . . |   lnk(e) = l
. . . | . . . |   ⇒
. . . | . . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                              kind(e)))
. . . | . . . 26. ∀e:E.
. . . | . . . |   loc(e) = source(l)
. . . | . . . |   ⇒
. . . | . . . |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . . 27. vartype(source(l);x) ⊆r A
. . . | . . . 28. ∀e:E.
. . . | . . . |   loc(e) = source(l)
. . . | . . . |   ⇒
. . . | . . . |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . 30. e : E
. . . | . . . 31. loc(e) = source(l)
. . . | . . . 32. kind(e) = k
. . . | . . . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . 34. ∀e':E.
. . . | . . . |   (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . |   =
. . . | . . . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . 38. f((x when e),val(e)) ∈ T List
. . . | . . . ⊢  map((λp.2of(p)) o (λe'.<tag(e'),val(e')>);L)
. . . | . . . |  =
. . . | . . . |  f((x when e),val(e))  by RW0 "map-map<" 0
. . . | . . . |
. . . | . . . 1. x : Id
. . . | . . . 2. tg : Id
. . . | . . . 3. k : Knd
. . . | . . . 4. l : IdLnk
. . . | . . . 5. T : Type
. . . | . . . 6. A : Type
. . . | . . . 7. B : Type
. . . | . . . 8. f : A→B→(T List)
. . . | . . . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . 10. @source(l): ma-single-sends(x : A;
. . . | . . . |                              k : B ⊕ rcv(l; tg) : T;
```

```
. . . | . . . |                                k;
. . . | . . . |                                l;
. . . | . . . |                                [<tg,λs,v. f(s(x),v)>])
. . . | . . . |   ∈ Dsys
. . . | . . . 11. ∀D:Dsys.
. . . | . . . |   @source(l): ma-single-sends(x : A;
. . . | . . . |                                k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                                k;
. . . | . . . |                                l;
. . . | . . . |                                [<tg,λs,v. f(s(x),v)>]
. . . | . . . |                                ) ⊆ D
. . . | . . . |   ⇒
. . . | . . . |   D
. . . | . . . |   realizes es.(∀x@0:Id.
. . . | . . . |                vartype(source(l);x@0) ⊆r ...(x@0)?Top)
. . . | . . . |               & (∀e:E.
. . . | . . . |                  loc(e) = source(l)
. . . | . . . |                  ⇒
. . . | . . . |                  (valtype(e) ⊆r ma-valtype
. . . | . . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . |               & (∀e:E.
. . . | . . . |                  isrcv(e)
. . . | . . . |                  ⇒
. . . | . . . |                  lnk(e) = l
. . . | . . . |                  ⇒
. . . | . . . |                  (valtype(e) ⊆r ma-valtype
. . . | . . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . |               & (∀e:E.
. . . | . . . |                  loc(e) = source(l)
. . . | . . . |                  ⇒
. . . | . . . |                  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);[<tg
. . . | . . . |                                        ,λs,v. f
. . . | . . . |                                               (s(x)
. . . | . . . |                                               ,v)>])))
. . . | . . . 12. @source(l): ma-single-sends(x : A;
. . . | . . . |                                k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                                k;
. . . | . . . |                                l;
. . . | . . . |                                [<tg,λs,v. f(s(x),v)>])
```

```
. . . | . . . |   ∈ Dsys
. . . | . . . 13. D : Dsys
. . . | . . . 14. @source(l): ma-single-sends(x : A;
. . . | . . . |                               k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                               k;
. . . | . . . |                               l;
. . . | . . . |                               [<tg,λs,v. f(s(x),v)>]
. . . | . . . |                               ) ⊆ D
. . . | . . . 15. ∀D':Dsys.
. . . | . . . |   D ⊆ D'
. . . | . . . |   ⇒
. . . | . . . |   (∀w:World, p:FairFifo.
. . . | . . . |    PossibleWorld(D';w)
. . . | . . . |    ⇒
. . . | . . . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . |    & (∀e:E.
. . . | . . . |       loc(e) = source(l)
. . . | . . . |       ⇒
. . . | . . . |       (valtype(e) ⊆r ma-valtype
. . . | . . . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . |    & (∀e:E.
. . . | . . . |       isrcv(e)
. . . | . . . |       ⇒
. . . | . . . |       lnk(e) = l
. . . | . . . |       ⇒
. . . | . . . |       (valtype(e) ⊆r ma-valtype
. . . | . . . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . |    & (∀e:E.
. . . | . . . |       loc(e) = source(l)
. . . | . . . |       ⇒
. . . | . . . |       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);[<tg,λs,v. f(s(x),v)>])))))
. . . | . . . 16. D' : Dsys
. . . | . . . 17. D ⊆ D'
. . . | . . . 18. ∀w:World, p:FairFifo.
. . . | . . . |   PossibleWorld(D';w)
. . . | . . . |   ⇒
. . . | . . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . |   & (∀e:E.
. . . | . . . |      loc(e) = source(l)
. . . | . . . |      ⇒
. . . | . . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                                kind(e))))
. . . | . . . |   & (∀e:E.
. . . | . . . |      isrcv(e)
```

```
. . . | . . . |      ⇒
. . . | . . . |      lnk(e) = l
. . . | . . . |      ⇒
. . . | . . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                                kind(e))))
. . . | . . . |   & (∀e:E.
. . . | . . . |      loc(e) = source(l)
. . . | . . . |      ⇒
. . . | . . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . . 19. w : World
. . . | . . . 20. ∀p:FairFifo.
. . . | . . . |   PossibleWorld(D';w)
. . . | . . . |   ⇒
. . . | . . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . |   & (∀e:E.
. . . | . . . |      loc(e) = source(l)
. . . | . . . |      ⇒
. . . | . . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                                kind(e))))
. . . | . . . |   & (∀e:E.
. . . | . . . |      isrcv(e)
. . . | . . . |      ⇒
. . . | . . . |      lnk(e) = l
. . . | . . . |      ⇒
. . . | . . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                                kind(e))))
. . . | . . . |   & (∀e:E.
. . . | . . . |      loc(e) = source(l)
. . . | . . . |      ⇒
. . . | . . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . . 21. p : FairFifo
. . . | . . . 22. PossibleWorld(D';w)
. . . | . . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
```

```
. . . | . . . 24. ∀e:E.
. . . | . . . |   loc(e) = source(l)
. . . | . . . |   ⇒
. . . | . . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                            kind(e)))
. . . | . . . 25. ∀e:E.
. . . | . . . |   isrcv(e)
. . . | . . . |   ⇒
. . . | . . . |   lnk(e) = l
. . . | . . . |   ⇒
. . . | . . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . |                            kind(e)))
. . . | . . . 26. ∀e:E.
. . . | . . . |   loc(e) = source(l)
. . . | . . . |   ⇒
. . . | . . . |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . . 27. vartype(source(l);x) ⊆r A
. . . | . . . 28. ∀e:E.
. . . | . . . |   loc(e) = source(l)
. . . | . . . |   ⇒
. . . | . . . |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . 30. e : E
. . . | . . . 31. loc(e) = source(l)
. . . | . . . 32. kind(e) = k
. . . | . . . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . 34. ∀e':E.
. . . | . . . |   (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . |   =
. . . | . . . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . 38. f((x when e),val(e)) ∈ T List
. . . | . . . ⊢  map(λp.2of(p);map(λe'.<tag(e'),val(e')>;L))
. . . | . . . |  =
. . . | . . . |  f((x when e),val(e))
. . . | . . . | by Subst
. . . | . . . |    ((f((x when e),val(e))) ∼ map(λp.2of(p);map(λx@0.
. . . | . . . |     <tg,x@0>;f((x when e),val(e)))))
. . . | . . . |    0
. . . | . . . |
. . . | . . . |\
. . . | . . . | .....equality .....
```

```
. . . | . . . | 1. x : Id
. . . | . . . | 2. tg : Id
. . . | . . . | 3. k : Knd
. . . | . . . | 4. l : IdLnk
. . . | . . . | 5. T : Type
. . . | . . . | 6. A : Type
. . . | . . . | 7. B : Type
. . . | . . . | 8. f : A→B→(T List)
. . . | . . . | 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . | 10. @source(l): ma-single-sends(x : A;
. . . | . . . | |                              k : B ⊕ rcv
. . . | . . . | |                              (l; tg) : T;
. . . | . . . | |                              k;
. . . | . . . | |                              l;
. . . | . . . | |                              [<tg,λs,v. f(s(x),v)>]
. . . | . . . | |                              )
. . . | . . . | |   ∈ Dsys
. . . | . . . | 11. ∀D:Dsys.
. . . | . . . | |   @source(l): ma-single-sends(x : A;
. . . | . . . | |                              k : B ⊕ rcv
. . . | . . . | |                              (l; tg) : T;
. . . | . . . | |                              k;
. . . | . . . | |                              l;
. . . | . . . | |                              [<tg,λs,v. f(s(x),v)>]
. . . | . . . | |                              ) ⊆ D
. . . | . . . | |   ⇒
. . . | . . . | |   D
. . . | . . . | |   realizes es.(∀x@0:Id.
. . . | . . . | |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . | |              & (∀e:E.
. . . | . . . | |                 loc(e) = source(l)
. . . | . . . | |                 ⇒
. . . | . . . | |                 (valtype(e) ⊆r ma-valtype
. . . | . . . | |                  (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . | |                   )))
. . . | . . . | |              & (∀e:E.
. . . | . . . | |                 isrcv(e)
. . . | . . . | |                 ⇒
. . . | . . . | |                 lnk(e) = l
. . . | . . . | |                 ⇒
. . . | . . . | |                 (valtype(e) ⊆r ma-valtype
. . . | . . . | |                  (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . | |                   )))
. . . | . . . | |              & (∀e:E.
. . . | . . . | |                 loc(e) = source(l)
. . . | . . . | |                 ⇒
. . . | . . . | |                 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);[<tg
. . . | . . . | |                                       ,λs,v.
. . . | . . . | |                                        f
. . . | . . . | |                                        (s(x)
. . . | . . . | |                                        ,v)>])))
. . . | . . . | 12. @source(l): ma-single-sends(x : A;
. . . | . . . | |                                 k : B ⊕ rcv
. . . | . . . | |                                 (l; tg) : T;
. . . | . . . | |                                 k;
. . . | . . . | |                                 l;
. . . | . . . | |                                 [<tg,λs,v. f(s(x),v)>]
. . . | . . . | |                                 )
. . . | . . . | |   ∈ Dsys
. . . | . . . | 13. D : Dsys
. . . | . . . | 14. @source(l): ma-single-sends(x : A;
. . . | . . . | |                                 k : B ⊕ rcv
. . . | . . . | |                                 (l; tg) : T;
. . . | . . . | |                                 k;
. . . | . . . | |                                 l;
. . . | . . . | |                                 [<tg,λs,v. f(s(x),v)>]
. . . | . . . | |                                 ) ⊆ D
. . . | . . . | 15. ∀D':Dsys.
. . . | . . . | |   D ⊆ D'
. . . | . . . | |   ⇒
. . . | . . . | |   (∀w:World, p:FairFifo.
. . . | . . . | |    PossibleWorld(D';w)
. . . | . . . | |    ⇒
. . . | . . . | |    (∀x@0:Id.
. . . | . . . | |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . | |    & (∀e:E.
. . . | . . . | |       loc(e) = source(l)
. . . | . . . | |       ⇒
. . . | . . . | |       (valtype(e) ⊆r ma-valtype
. . . | . . . | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | |    & (∀e:E.
. . . | . . . | |       isrcv(e)
. . . | . . . | |       ⇒
. . . | . . . | |       lnk(e) = l
. . . | . . . | |       ⇒
. . . | . . . | |       (valtype(e) ⊆r ma-valtype
. . . | . . . | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | |    & (∀e:E.
. . . | . . . | |       loc(e) = source(l)
. . . | . . . | |       ⇒
. . . | . . . | |       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);[<tg
. . . | . . . | |                              ,λs,v. f(s(x),v)>])))
. . . | . . . | 16. D' : Dsys
. . . | . . . | 17. D ⊆ D'
. . . | . . . | 18. ∀w:World, p:FairFifo.
. . . | . . . | |   PossibleWorld(D';w)
. . . | . . . | |   ⇒
. . . | . . . | |   (∀x@0:Id.
. . . | . . . | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . | |   & (∀e:E.
. . . | . . . | |      loc(e) = source(l)
. . . | . . . | |      ⇒
. . . | . . . | |      (valtype(e) ⊆r ma-valtype
. . . | . . . | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | |   & (∀e:E.
. . . | . . . | |      isrcv(e)
. . . | . . . | |      ⇒
. . . | . . . | |      lnk(e) = l
. . . | . . . | |      ⇒
. . . | . . . | |      (valtype(e) ⊆r ma-valtype
. . . | . . . | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | |   & (∀e:E.
. . . | . . . | |      loc(e) = source(l)
. . . | . . . | |      ⇒
. . . | . . . | |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . . | 19. w : World
. . . | . . . | 20. ∀p:FairFifo.
. . . | . . . | |   PossibleWorld(D';w)
. . . | . . . | |   ⇒
. . . | . . . | |   (∀x@0:Id.
. . . | . . . | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . | |   & (∀e:E.
. . . | . . . | |      loc(e) = source(l)
. . . | . . . | |      ⇒
```

```
. . . | . . . | |      (valtype(e) ⊆r ma-valtype
. . . | . . . | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | |   & (∀e:E.
. . . | . . . | |      isrcv(e)
. . . | . . . | |      ⇒
. . . | . . . | |      lnk(e) = l
. . . | . . . | |      ⇒
. . . | . . . | |      (valtype(e) ⊆r ma-valtype
. . . | . . . | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | |   & (∀e:E.
. . . | . . . | |      loc(e) = source(l)
. . . | . . . | |      ⇒
. . . | . . . | |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . . | 21. p : FairFifo
. . . | . . . | 22. PossibleWorld(D';w)
. . . | . . . | 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . | 24. ∀e:E.
. . . | . . . | |   loc(e) = source(l)
. . . | . . . | |   ⇒
. . . | . . . | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . | |                             kind(e)))
. . . | . . . | 25. ∀e:E.
. . . | . . . | |   isrcv(e)
. . . | . . . | |   ⇒
. . . | . . . | |   lnk(e) = l
. . . | . . . | |   ⇒
. . . | . . . | |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . | |                             kind(e)))
. . . | . . . | 26. ∀e:E.
. . . | . . . | |   loc(e) = source(l)
. . . | . . . | |   ⇒
. . . | . . . | |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . . | 27. vartype(source(l);x) ⊆r A
```

```
. . . | . . . | 28. ∀e:E.
. . . | . . . | |   loc(e) = source(l)
. . . | . . . | |   ⇒
. . . | . . . | |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . | 30. e : E
. . . | . . . | 31. loc(e) = source(l)
. . . | . . . | 32. kind(e) = k
. . . | . . . | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . | 34. ∀e':E.
. . . | . . . | |   (e' ∈ L)
. . . | . . . | |   ⟺
. . . | . . . | |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . | |   =
. . . | . . . | |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . | 38. f((x when e),val(e)) ∈ T List
. . . | . . . | ⊢  (f((x when e),val(e))) ∼ map(λp.2of(p);map(λx@0.
. . . | . . . | |  <tg,x@0>;f((x when e),val(e))))
. . . | . . . | | by RW0

. . . | . . . | |    Thm* ∀as:Top List, f,g:Top.
. . . | . . . | |         map(g;map(f;as)) ∼ map(g o f;as)
. . . | . . . | |    {map_DASH_map}
. . . | . . . | |    0
. . . | . . . | |    THENA
. . . | . . . | |    Try (Complete Auto)
. . . | . . . | |
. . . | . . . | |\
. . . | . . . | | .....wf .....
. . . | . . . | | 1. x : Id
. . . | . . . | | 2. tg : Id
. . . | . . . | | 3. k : Knd
. . . | . . . | | 4. l : IdLnk
. . . | . . . | | 5. T : Type
. . . | . . . | | 6. A : Type
. . . | . . . | | 7. B : Type
. . . | . . . | | 8. f : A→B→(T List)
. . . | . . . | | 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . | | 10. @source(l): ma-single-sends(x : A;
. . . | . . . | |                                 k : B ⊕ rcv
. . . | . . . | |                                 (l; tg) : T;
. . . | . . . | |                                 k;
. . . | . . . | |                                 l;
. . . | . . . | |                                 [<tg
. . . | . . . | |                                  ,λs,v. f(s(x),v)>])
. . . | . . . | |     ∈ Dsys
. . . | . . . | | 11. ∀D:Dsys.
. . . | . . . | |     @source(l): ma-single-sends(x : A;
. . . | . . . | |                                 k : B ⊕ rcv
. . . | . . . | |                                 (l; tg) : T;
. . . | . . . | |                                 k;
. . . | . . . | |                                 l;
. . . | . . . | |                                 [<tg
```

```
. . . | . . . | |                                  ,λs,v. f(s(x),v)>]
. . . | . . . | |                                ) ⊆ D
. . . | . . . | |     ⇒
. . . | . . . | |     D
. . . | . . . | |     realizes es.(∀x@0:Id.
. . . | . . . | |                   vartype(source(l);x@0) ⊆r ...)
. . . | . . . | |                 & (∀e:E.
. . . | . . . | |                    loc(e) = source(l)
. . . | . . . | |                    ⇒
. . . | . . . | |                    (valtype(e) ⊆r ma-valtype
. . . | . . . | |                     (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . | |                      )))
. . . | . . . | |                 & (∀e:E.
. . . | . . . | |                    isrcv(e)
. . . | . . . | |                    ⇒
. . . | . . . | |                    lnk(e) = l
. . . | . . . | |                    ⇒
. . . | . . . | |                    (valtype(e) ⊆r ma-valtype
. . . | . . . | |                     (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . | |                      )))
. . . | . . . | |                 & (∀e:E.
. . . | . . . | |                    loc(e) = source(l)
. . . | . . . | |                    ⇒
. . . | . . . | |                    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);[<tg
. . . | . . . | |                                          ,λs,v.
. . . | . . . | |                                           f
. . . | . . . | |                                           (s(x)
. . . | . . . | |                                           ,v)>])))
. . . | . . . | | 12. @source(l): ma-single-sends(x : A;
. . . | . . . | |                                 k : B ⊕ rcv
. . . | . . . | |                                 (l; tg) : T;
. . . | . . . | |                                 k;
. . . | . . . | |                                 l;
. . . | . . . | |                                 [<tg
. . . | . . . | |                                  ,λs,v. f(s(x),v)>])
. . . | . . . | |     ∈ Dsys
. . . | . . . | | 13. D : Dsys
```

```
. . . | . . . | | 14. @source(l): ma-single-sends(x : A;
. . . | . . . | |                                k : B ⊕ rcv
. . . | . . . | |                                (l; tg) : T;
. . . | . . . | |                                k;
. . . | . . . | |                                l;
. . . | . . . | |                                [<tg
. . . | . . . | |                                 ,λs,v. f(s(x),v)>]
. . . | . . . | |                                ) ⊆ D
. . . | . . . | | 15. ∀D':Dsys.
. . . | . . . | |     D ⊆ D'
. . . | . . . | |     ⇒
. . . | . . . | |     (∀w:World, p:FairFifo.
. . . | . . . | |      PossibleWorld(D';w)
. . . | . . . | |      ⇒
. . . | . . . | |      (∀x@0:Id.
. . . | . . . | |       vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . | |      & (∀e:E.
. . . | . . . | |         loc(e) = source(l)
. . . | . . . | |         ⇒
. . . | . . . | |         (valtype(e) ⊆r ma-valtype
. . . | . . . | |          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | |      & (∀e:E.
. . . | . . . | |         isrcv(e)
. . . | . . . | |         ⇒
. . . | . . . | |         lnk(e) = l
. . . | . . . | |         ⇒
. . . | . . . | |         (valtype(e) ⊆r ma-valtype
. . . | . . . | |          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | |      & (∀e:E.
. . . | . . . | |         loc(e) = source(l)
. . . | . . . | |         ⇒
. . . | . . . | |         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);[<tg
. . . | . . . | |                                 ,λs,v. f(s(x),v)>])))))
. . . | . . . | | 16. D' : Dsys
. . . | . . . | | 17. D ⊆ D'
. . . | . . . | | 18. ∀w:World, p:FairFifo.
. . . | . . . | |     PossibleWorld(D';w)
. . . | . . . | |     ⇒
. . . | . . . | |     (∀x@0:Id.
. . . | . . . | |      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . | |     & (∀e:E.
. . . | . . . | |        loc(e) = source(l)
. . . | . . . | |        ⇒
```

```
. . . | . . . | |        (valtype(e) ⊆r ma-valtype
. . . | . . . | |         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | |     & (∀e:E.
. . . | . . . | |        isrcv(e)
. . . | . . . | |        ⇒
. . . | . . . | |        lnk(e) = l
. . . | . . . | |        ⇒
. . . | . . . | |        (valtype(e) ⊆r ma-valtype
. . . | . . . | |         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | |     & (∀e:E.
. . . | . . . | |        loc(e) = source(l)
. . . | . . . | |        ⇒
. . . | . . . | |        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);[<tg
. . . | . . . | |                                ,λs,v. f(s(x),v)>])))
. . . | . . . | | 19. w : World
. . . | . . . | | 20. ∀p:FairFifo.
. . . | . . . | |     PossibleWorld(D';w)
. . . | . . . | |     ⇒
. . . | . . . | |     (∀x@0:Id.
. . . | . . . | |      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . | |     & (∀e:E.
. . . | . . . | |        loc(e) = source(l)
. . . | . . . | |        ⇒
. . . | . . . | |        (valtype(e) ⊆r ma-valtype
. . . | . . . | |         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | |     & (∀e:E.
. . . | . . . | |        isrcv(e)
. . . | . . . | |        ⇒
. . . | . . . | |        lnk(e) = l
. . . | . . . | |        ⇒
. . . | . . . | |        (valtype(e) ⊆r ma-valtype
. . . | . . . | |         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | |     & (∀e:E.
. . . | . . . | |        loc(e) = source(l)
. . . | . . . | |        ⇒
. . . | . . . | |        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);[<tg
. . . | . . . | |                              ,λs,v. f(s(x),v)>])))
. . . | . . . | | 21. p : FairFifo
. . . | . . . | | 22. PossibleWorld(D';w)
. . . | . . . | | 23. ∀x@0:Id.
. . . | . . . | |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . | | 24. ∀e:E.
. . . | . . . | |     loc(e) = source(l)
. . . | . . . | |     ⇒
. . . | . . . | |     (valtype(e) ⊆r ma-valtype
. . . | . . . | |      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . | | 25. ∀e:E.
. . . | . . . | |     isrcv(e)
. . . | . . . | |     ⇒
. . . | . . . | |     lnk(e) = l
. . . | . . . | |     ⇒
. . . | . . . | |     (valtype(e) ⊆r ma-valtype
. . . | . . . | |      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . | | 26. ∀e:E.
. . . | . . . | |     loc(e) = source(l)
. . . | . . . | |     ⇒
. . . | . . . | |     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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . . | | 27. vartype(source(l);x) ⊆r A
. . . | . . . | | 28. ∀e:E.
. . . | . . . | |     loc(e) = source(l)
. . . | . . . | |     ⇒
. . . | . . . | |     kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . | | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . | | 30. e : E
. . . | . . . | | 31. loc(e) = source(l)
. . . | . . . | | 32. kind(e) = k
. . . | . . . | | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . | | 34. ∀e':E.
. . . | . . . | |     (e' ∈ L)
. . . | . . . | |     ⟺
. . . | . . . | |     isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . | | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . | | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . | |     =
. . . | . . . | |     map(λx@0.<tg,x@0>;f((x when e),val(e)))
```

```
. . . | . . . | | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . | | 38. f((x when e),val(e)) ∈ T List
. . . | . . . | | ⊢  f((x when e),val(e)) ∈ Top List
. . . | . . . | |   by DoSubsume THEN UseSubtypeRelLemmas
. . . | . . . | |      THEN
. . . | . . . | |      Analyze 0
. . . | . . . | |
. . . | . . . | .\
. . . | . . . | . 1. x : Id
. . . | . . . | . 2. tg : Id
. . . | . . . | . 3. k : Knd
. . . | . . . | . 4. l : IdLnk
. . . | . . . | . 5. T : Type
. . . | . . . | . 6. A : Type
. . . | . . . | . 7. B : Type
. . . | . . . | . 8. f : A→B→(T List)
. . . | . . . | . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . | . 10. @source(l): ma-single-sends(x : A;
. . . | . . . | . |                               k : B ⊕ rcv
. . . | . . . | . |                               (l; tg) : T;
. . . | . . . | . |                               k;
. . . | . . . | . |                               l;
. . . | . . . | . |                               [<tg
. . . | . . . | . |                                ,λs,v. f(s(x),v)>])
. . . | . . . | . |   ∈ Dsys
. . . | . . . | . 11. ∀D:Dsys.
. . . | . . . | . |   @source(l): ma-single-sends(x : A;
. . . | . . . | . |                               k : B ⊕ rcv
. . . | . . . | . |                               (l; tg) : T;
. . . | . . . | . |                               k;
. . . | . . . | . |                               l;
. . . | . . . | . |                               [<tg
. . . | . . . | . |                                ,λs,v. f(s(x),v)>]
. . . | . . . | . |                               ) ⊆ D
. . . | . . . | . |   ⇒
. . . | . . . | . |   D
. . . | . . . | . |   realizes es.(∀x@0:Id.
. . . | . . . | . |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . | . |               & (∀e:E.
. . . | . . . | . |                  loc(e) = source(l)
. . . | . . . | . |                  ⇒
. . . | . . . | . |                  (valtype(e) ⊆r ma-valtype
. . . | . . . | . |                   (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . | . |                    )))
. . . | . . . | . |               & (∀e:E.
. . . | . . . | . |                  isrcv(e)
. . . | . . . | . |                  ⇒
. . . | . . . | . |                  lnk(e) = l
. . . | . . . | . |                  ⇒
. . . | . . . | . |                  (valtype(e) ⊆r ma-valtype
. . . | . . . | . |                   (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . | . |                    )))
. . . | . . . | . |               & (∀e:E.
. . . | . . . | . |                  loc(e) = source(l)
. . . | . . . | . |                  ⇒
```

```
. . . | . . . | . |                  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);[<tg
. . . | . . . | . |                                        ,λs,v.
. . . | . . . | . |                                         f
. . . | . . . | . |                                         (s(x)
. . . | . . . | . |                                         ,v)>])))
. . . | . . . | . 12. @source(l): ma-single-sends(x : A;
. . . | . . . | . |                                 k : B ⊕ rcv
. . . | . . . | . |                                 (l; tg) : T;
. . . | . . . | . |                                 k;
. . . | . . . | . |                                 l;
. . . | . . . | . |                                 [<tg
. . . | . . . | . |                                  ,λs,v. f(s(x),v)>])
. . . | . . . | . |   ∈ Dsys
. . . | . . . | . 13. D : Dsys
. . . | . . . | . 14. @source(l): ma-single-sends(x : A;
. . . | . . . | . |                                 k : B ⊕ rcv
. . . | . . . | . |                                 (l; tg) : T;
. . . | . . . | . |                                 k;
. . . | . . . | . |                                 l;
. . . | . . . | . |                                 [<tg
. . . | . . . | . |                                  ,λs,v. f(s(x),v)>]
. . . | . . . | . |                                 ) ⊆ D
. . . | . . . | . 15. ∀D':Dsys.
. . . | . . . | . |   D ⊆ D'
. . . | . . . | . |   ⇒
. . . | . . . | . |   (∀w:World, p:FairFifo.
. . . | . . . | . |    PossibleWorld(D';w)
. . . | . . . | . |    ⇒
. . . | . . . | . |    (∀x@0:Id.
. . . | . . . | . |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . | . |    & (∀e:E.
. . . | . . . | . |       loc(e) = source(l)
. . . | . . . | . |       ⇒
. . . | . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | . |    & (∀e:E.
. . . | . . . | . |       isrcv(e)
```

```
. . . | . . . | . |       ⇒
. . . | . . . | . |       lnk(e) = l
. . . | . . . | . |       ⇒
. . . | . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | . |    & (∀e:E.
. . . | . . . | . |       loc(e) = source(l)
. . . | . . . | . |       ⇒
. . . | . . . | . |       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);[<tg
. . . | . . . | . |                              ,λs,v. f(s(x),v)>])))))
. . . | . . . | . 16. D' : Dsys
. . . | . . . | . 17. D ⊆ D'
. . . | . . . | . 18. ∀w:World, p:FairFifo.
. . . | . . . | . |   PossibleWorld(D';w)
. . . | . . . | . |   ⇒
. . . | . . . | . |   (∀x@0:Id.
. . . | . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . | . |   & (∀e:E.
. . . | . . . | . |      loc(e) = source(l)
. . . | . . . | . |      ⇒
. . . | . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | . |   & (∀e:E.
. . . | . . . | . |      isrcv(e)
. . . | . . . | . |      ⇒
. . . | . . . | . |      lnk(e) = l
. . . | . . . | . |      ⇒
. . . | . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | . |   & (∀e:E.
. . . | . . . | . |      loc(e) = source(l)
. . . | . . . | . |      ⇒
. . . | . . . | . |      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);[<tg
. . . | . . . | . |                              ,λs,v. f(s(x),v)>])))
. . . | . . . | . 19. w : World
. . . | . . . | . 20. ∀p:FairFifo.
. . . | . . . | . |   PossibleWorld(D';w)
. . . | . . . | . |   ⇒
. . . | . . . | . |   (∀x@0:Id.
. . . | . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . | . |   & (∀e:E.
. . . | . . . | . |      loc(e) = source(l)
. . . | . . . | . |      ⇒
. . . | . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | . |   & (∀e:E.
. . . | . . . | . |      isrcv(e)
. . . | . . . | . |      ⇒
. . . | . . . | . |      lnk(e) = l
. . . | . . . | . |      ⇒
. . . | . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | . |   & (∀e:E.
. . . | . . . | . |      loc(e) = source(l)
. . . | . . . | . |      ⇒
. . . | . . . | . |      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);[<tg
. . . | . . . | . |                              ,λs,v. f(s(x),v)>])))
. . . | . . . | . 21. p : FairFifo
. . . | . . . | . 22. PossibleWorld(D';w)
. . . | . . . | . 23. ∀x@0:Id.
. . . | . . . | . |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . | . 24. ∀e:E.
. . . | . . . | . |   loc(e) = source(l)
. . . | . . . | . |   ⇒
. . . | . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . | . 25. ∀e:E.
. . . | . . . | . |   isrcv(e)
. . . | . . . | . |   ⇒
. . . | . . . | . |   lnk(e) = l
. . . | . . . | . |   ⇒
. . . | . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . | . 26. ∀e:E.
```

```
. . . | . . . | . |   loc(e) = source(l)
. . . | . . . | . |   ⇒
. . . | . . . | . |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . . | . 27. vartype(source(l);x) ⊆r A
. . . | . . . | . 28. ∀e:E.
. . . | . . . | . |   loc(e) = source(l)
. . . | . . . | . |   ⇒
. . . | . . . | . |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . | . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . | . 30. e : E
. . . | . . . | . 31. loc(e) = source(l)
. . . | . . . | . 32. kind(e) = k
. . . | . . . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . | . 34. ∀e':E.
. . . | . . . | . |   (e' ∈ L)
. . . | . . . | . |   ⟺
. . . | . . . | . |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . | . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . | . |   =
. . . | . . . | . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . | . 38. f((x when e),val(e)) ∈ T List
. . . | . . . | . ⊢  (f((x when e),val(e))) ∼ map((λp.2of(p))
. . . | . . . | . |   o (λx@0.<tg,x@0>);f((x when e),val(e)))
. . . | . . . | . | by Unfold ‘compose‘ 0 THEN Reduce 0
. . . | . . . | . |
. . . | . . . | . 1. x : Id
. . . | . . . | . 2. tg : Id
. . . | . . . | . 3. k : Knd
. . . | . . . | . 4. l : IdLnk
. . . | . . . | . 5. T : Type
. . . | . . . | . 6. A : Type
. . . | . . . | . 7. B : Type
. . . | . . . | . 8. f : A→B→(T List)
. . . | . . . | . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . | . 10. @source(l): ma-single-sends(x : A;
. . . | . . . | .                                k : B ⊕ rcv
. . . | . . . | .                                (l; tg) : T;
. . . | . . . | .                                k;
. . . | . . . | .                                l;
. . . | . . . | .                                [<tg
. . . | . . . | .                                 ,λs,v. f(s(x),v)>])
. . . | . . . | .     ∈ Dsys
```

```
. . . | . . . | . 11. ∀D:Dsys.
. . . | . . . | .     @source(l): ma-single-sends(x : A;
. . . | . . . | .                                k : B ⊕ rcv
. . . | . . . | .                                (l; tg) : T;
. . . | . . . | .                                k;
. . . | . . . | .                                l;
. . . | . . . | .                                [<tg
. . . | . . . | .                                 ,λs,v. f(s(x),v)>]
. . . | . . . | .                                ) ⊆ D
. . . | . . . | .     ⇒
. . . | . . . | .     D
. . . | . . . | .     realizes es.(∀x@0:Id.
. . . | . . . | .                  vartype(source(l);x@0) ⊆r ...)
. . . | . . . | .                & (∀e:E.
. . . | . . . | .                   loc(e) = source(l)
. . . | . . . | .                   ⇒
. . . | . . . | .                   (valtype(e) ⊆r ma-valtype
. . . | . . . | .                    (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . | .                     )))
. . . | . . . | .                & (∀e:E.
. . . | . . . | .                   isrcv(e)
. . . | . . . | .                   ⇒
. . . | . . . | .                   lnk(e) = l
. . . | . . . | .                   ⇒
. . . | . . . | .                   (valtype(e) ⊆r ma-valtype
. . . | . . . | .                    (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . | .                     )))
. . . | . . . | .                & (∀e:E.
. . . | . . . | .                   loc(e) = source(l)
. . . | . . . | .                   ⇒
. . . | . . . | .                   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);[<tg
. . . | . . . | .                                         ,λs,v.
. . . | . . . | .                                          f
. . . | . . . | .                                          (s(x)
. . . | . . . | .                                          ,v)>])))
. . . | . . . | . 12. @source(l): ma-single-sends(x : A;
. . . | . . . | .                                k : B ⊕ rcv
```

```
. . . | . . . | .                                  (l; tg) : T;
. . . | . . . | .                                  k;
. . . | . . . | .                                  l;
. . . | . . . | .                                  [<tg
. . . | . . . | .                                   ,λs,v. f(s(x),v)>])
. . . | . . . | .     ∈ Dsys
. . . | . . . | . 13. D : Dsys
. . . | . . . | . 14. @source(l): ma-single-sends(x : A;
. . . | . . . | .                                  k : B ⊕ rcv
. . . | . . . | .                                  (l; tg) : T;
. . . | . . . | .                                  k;
. . . | . . . | .                                  l;
. . . | . . . | .                                  [<tg
. . . | . . . | .                                   ,λs,v. f(s(x),v)>]
. . . | . . . | .                                  ) ⊆ D
. . . | . . . | . 15. ∀D':Dsys.
. . . | . . . | .     D ⊆ D'

. . . | . . . | .     ⇒
. . . | . . . | .     (∀w:World, p:FairFifo.
. . . | . . . | .      PossibleWorld(D';w)
. . . | . . . | .      ⇒
. . . | . . . | .      (∀x@0:Id.
. . . | . . . | .       vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . | .      & (∀e:E.
. . . | . . . | .         loc(e) = source(l)
. . . | . . . | .         ⇒
. . . | . . . | .         (valtype(e) ⊆r ma-valtype
. . . | . . . | .          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | .      & (∀e:E.
. . . | . . . | .         isrcv(e)
. . . | . . . | .         ⇒
. . . | . . . | .         lnk(e) = l
. . . | . . . | .         ⇒
. . . | . . . | .         (valtype(e) ⊆r ma-valtype
. . . | . . . | .          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | .      & (∀e:E.
. . . | . . . | .         loc(e) = source(l)
. . . | . . . | .         ⇒
. . . | . . . | .         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);[<tg
. . . | . . . | .                                ,λs,v. f(s(x),v)>])))))
. . . | . . . | . 16. D' : Dsys
. . . | . . . | . 17. D ⊆ D'
. . . | . . . | . 18. ∀w:World, p:FairFifo.
```

```
. . . | . . . | .     PossibleWorld(D';w)
. . . | . . . | .     ⇒
. . . | . . . | .     (∀x@0:Id.
. . . | . . . | .      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . | .     & (∀e:E.
. . . | . . . | .        loc(e) = source(l)
. . . | . . . | .        ⇒
. . . | . . . | .        (valtype(e) ⊆r ma-valtype
. . . | . . . | .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | .     & (∀e:E.
. . . | . . . | .        isrcv(e)
. . . | . . . | .        ⇒
. . . | . . . | .        lnk(e) = l
. . . | . . . | .        ⇒
. . . | . . . | .        (valtype(e) ⊆r ma-valtype
. . . | . . . | .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | .     & (∀e:E.
. . . | . . . | .        loc(e) = source(l)
. . . | . . . | .        ⇒
. . . | . . . | .        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);[<tg
. . . | . . . | .                              ,λs,v. f(s(x),v)>])))
. . . | . . . | . 19. w : World
. . . | . . . | . 20. ∀p:FairFifo.
. . . | . . . | .     PossibleWorld(D';w)
. . . | . . . | .     ⇒
. . . | . . . | .     (∀x@0:Id.
. . . | . . . | .      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . | .     & (∀e:E.
. . . | . . . | .        loc(e) = source(l)
. . . | . . . | .        ⇒
. . . | . . . | .        (valtype(e) ⊆r ma-valtype
. . . | . . . | .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | .     & (∀e:E.
. . . | . . . | .        isrcv(e)
. . . | . . . | .        ⇒
. . . | . . . | .        lnk(e) = l
. . . | . . . | .        ⇒
. . . | . . . | .        (valtype(e) ⊆r ma-valtype
. . . | . . . | .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . | .     & (∀e:E.
. . . | . . . | .        loc(e) = source(l)
. . . | . . . | .        ⇒
. . . | . . . | .        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);[<tg
. . . | . . . | .                              ,λs,v. f(s(x),v)>])))
. . . | . . . | . 21. p : FairFifo
. . . | . . . | . 22. PossibleWorld(D';w)
. . . | . . . | . 23. ∀x@0:Id.
. . . | . . . | .     vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . | . 24. ∀e:E.
. . . | . . . | .     loc(e) = source(l)
. . . | . . . | .     ⇒
. . . | . . . | .     (valtype(e) ⊆r ma-valtype
. . . | . . . | .      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . | . 25. ∀e:E.
. . . | . . . | .     isrcv(e)
. . . | . . . | .     ⇒
. . . | . . . | .     lnk(e) = l
. . . | . . . | .     ⇒
. . . | . . . | .     (valtype(e) ⊆r ma-valtype
. . . | . . . | .      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . | . 26. ∀e:E.
. . . | . . . | .     loc(e) = source(l)
. . . | . . . | .     ⇒
. . . | . . . | .     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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . . | . 27. vartype(source(l);x) ⊆r A
. . . | . . . | . 28. ∀e:E.
. . . | . . . | .     loc(e) = source(l)
. . . | . . . | .     ⇒
. . . | . . . | .     kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . | . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . | . 30. e : E
. . . | . . . | . 31. loc(e) = source(l)
. . . | . . . | . 32. kind(e) = k
. . . | . . . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . | . 34. ∀e':E.
```

```
. . . | . . . | .     (e' ∈ L)
. . . | . . . | .     ⟺
. . . | . . . | .     isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . | . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . | .     =
. . . | . . . | .     map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . | . 38. f((x when e),val(e)) ∈ T List
. . . | . . . | . ⊢  (f((x when e),val(e))) ∼ map(λx.x;f
. . . | . . . | .                                   ((x when e)
. . . | . . . | .                                   ,val(e)))
. . . | . . . | .   by RW0
. . . | . . . | .      Thm* ∀L:Top List. map(λx.x;L) ∼ L
. . . | . . . | .      {map_DASH_id}
. . . | . . . | .      0
. . . | . . . | .      THEN
. . . | . . . | .      MaAuto
. . . | . . . |
. . . | . . . .\
. . . | . . . . 1. x : Id
. . . | . . . . 2. tg : Id
. . . | . . . . 3. k : Knd
. . . | . . . . 4. l : IdLnk
. . . | . . . . 5. T : Type
. . . | . . . . 6. A : Type
. . . | . . . . 7. B : Type
. . . | . . . . 8. f : A→B→(T List)
. . . | . . . . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . 10. @source(l): ma-single-sends(x : A;
. . . | . . . . |                              k : B ⊕ rcv
. . . | . . . . |                              (l; tg) : T;
. . . | . . . . |                              k;
. . . | . . . . |                              l;
. . . | . . . . |                              [<tg,λs,v. f(s(x),v)>]
. . . | . . . . |                              )
. . . | . . . . |   ∈ Dsys
. . . | . . . . 11. ∀D:Dsys.
. . . | . . . . |   @source(l): ma-single-sends(x : A;
. . . | . . . . |                              k : B ⊕ rcv
. . . | . . . . |                              (l; tg) : T;
. . . | . . . . |                              k;
. . . | . . . . |                              l;
. . . | . . . . |                              [<tg,λs,v. f(s(x),v)>]
. . . | . . . . |                              ) ⊆ D
. . . | . . . . |   ⇒
. . . | . . . . |   D
. . . | . . . . |   realizes es.(∀x@0:Id.
. . . | . . . . |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . . |               & (∀e:E.
. . . | . . . . |                  loc(e) = source(l)
. . . | . . . . |                  ⇒
. . . | . . . . |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . . |                    )))
```

```
. . . | . . . . |                & (∀e:E.
. . . | . . . . |                   isrcv(e)
. . . | . . . . |                   ⇒
. . . | . . . . |                   lnk(e) = l
. . . | . . . . |                   ⇒
. . . | . . . . |                   (valtype(e) ⊆r ma-valtype
. . . | . . . . |                    (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . . |                     )))
. . . | . . . . |                & (∀e:E.
. . . | . . . . |                   loc(e) = source(l)
. . . | . . . . |                   ⇒
. . . | . . . . |                   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);[<tg
. . . | . . . . |                                        ,λs,v.
. . . | . . . . |                                         f
. . . | . . . . |                                         (s(x)
. . . | . . . . |                                         ,v)>])))
. . . | . . . . 12. @source(l): ma-single-sends(x : A;
. . . | . . . . |                                k : B ⊕ rcv
. . . | . . . . |                                (l; tg) : T;
. . . | . . . . |                                k;
. . . | . . . . |                                l;
. . . | . . . . |                                [<tg,λs,v. f(s(x),v)>]
. . . | . . . . |                                )
. . . | . . . . |   ∈ Dsys
. . . | . . . . 13. D : Dsys
. . . | . . . . 14. @source(l): ma-single-sends(x : A;
. . . | . . . . |                                k : B ⊕ rcv
. . . | . . . . |                                (l; tg) : T;
. . . | . . . . |                                k;
. . . | . . . . |                                l;
. . . | . . . . |                                [<tg,λs,v. f(s(x),v)>]
. . . | . . . . |                                ) ⊆ D
. . . | . . . . 15. ∀D':Dsys.
. . . | . . . . |   D ⊆ D'
. . . | . . . . |   ⇒
. . . | . . . . |   (∀w:World, p:FairFifo.
. . . | . . . . |    PossibleWorld(D';w)
```

```
. . . | . . . . |    ⇒
. . . | . . . . |    (∀x@0:Id.
. . . | . . . . |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . |    & (∀e:E.
. . . | . . . . |       loc(e) = source(l)
. . . | . . . . |       ⇒
. . . | . . . . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |    & (∀e:E.
. . . | . . . . |       isrcv(e)
. . . | . . . . |       ⇒
. . . | . . . . |       lnk(e) = l
. . . | . . . . |       ⇒
. . . | . . . . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |    & (∀e:E.
. . . | . . . . |       loc(e) = source(l)
. . . | . . . . |       ⇒
. . . | . . . . |       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);[<tg
. . . | . . . . |                                ,λs,v. f(s(x),v)>])))))
. . . | . . . . 16. D' : Dsys
. . . | . . . . 17. D ⊆ D'
. . . | . . . . 18. ∀w:World, p:FairFifo.
. . . | . . . . |    PossibleWorld(D';w)
. . . | . . . . |    ⇒
. . . | . . . . |    (∀x@0:Id.
. . . | . . . . |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . |    & (∀e:E.
. . . | . . . . |       loc(e) = source(l)
. . . | . . . . |       ⇒
. . . | . . . . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |    & (∀e:E.
. . . | . . . . |       isrcv(e)
. . . | . . . . |       ⇒
. . . | . . . . |       lnk(e) = l
. . . | . . . . |       ⇒
. . . | . . . . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |    & (∀e:E.
. . . | . . . . |       loc(e) = source(l)
. . . | . . . . |       ⇒
. . . | . . . . |       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);[<tg,λs,v. f(s(x),v)>]))))
. . . | . . . . 19. w : World
. . . | . . . . 20. ∀p:FairFifo.
. . . | . . . . |   PossibleWorld(D';w)
. . . | . . . . |   ⇒
. . . | . . . . |   (∀x@0:Id.
. . . | . . . . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . |   & (∀e:E.
. . . | . . . . |      loc(e) = source(l)
. . . | . . . . |      ⇒
. . . | . . . . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |   & (∀e:E.
. . . | . . . . |      isrcv(e)
. . . | . . . . |      ⇒
. . . | . . . . |      lnk(e) = l
. . . | . . . . |      ⇒
. . . | . . . . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |   & (∀e:E.
. . . | . . . . |      loc(e) = source(l)
. . . | . . . . |      ⇒
. . . | . . . . |      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);[<tg,λs,v. f(s(x),v)>]))))
. . . | . . . . 21. p : FairFifo
. . . | . . . . 22. PossibleWorld(D';w)
. . . | . . . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . 24. ∀e:E.
. . . | . . . . |   loc(e) = source(l)
. . . | . . . . |   ⇒
. . . | . . . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . |                             kind(e)))
. . . | . . . . 25. ∀e:E.
. . . | . . . . |   isrcv(e)
. . . | . . . . |   ⇒
```

```
. . . | . . . . |   lnk(e) = l
. . . | . . . . |   ⇒
. . . | . . . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . |                             kind(e)))
. . . | . . . . 26. ∀e:E.
. . . | . . . . |   loc(e) = source(l)
. . . | . . . . |   ⇒
. . . | . . . . |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . . . 27. vartype(source(l);x) ⊆r A
. . . | . . . . 28. ∀e:E.
. . . | . . . . |   loc(e) = source(l)
. . . | . . . . |   ⇒
. . . | . . . . |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . 30. e : E
. . . | . . . . 31. loc(e) = source(l)
. . . | . . . . 32. kind(e) = k
. . . | . . . . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . 34. ∀e':E.
. . . | . . . . |   (e' ∈ L)
. . . | . . . . |   ⟺

. . . | . . . . |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . |   =
. . . | . . . . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . 38. f((x when e),val(e)) ∈ T List
. . . | . . . . ⊢  map(λp.2of(p);map(λe'.<tag(e'),val(e')>;L))
. . . | . . . . |  =
. . . | . . . . |  map(λp.2of(p);map(λx@0.<tg,x@0>;f
. . . | . . . . |                                 ((x when e)
. . . | . . . . |                                 ,val(e))))
. . . | . . . . | by Subst
. . . | . . . . |    (map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . |     =
. . . | . . . . |     map(λx@0.<tg,x@0>;f((x when e),val(e))))
. . . | . . . . |    0
. . . | . . . . |
. . . | . . . . .....equality .....
. . . | . . . . 1. x : Id
. . . | . . . . 2. tg : Id
. . . | . . . . 3. k : Knd
. . . | . . . . 4. l : IdLnk
```

```
. . . | . . . . 5. T : Type
. . . | . . . . 6. A : Type
. . . | . . . . 7. B : Type
. . . | . . . . 8. f : A→B→(T List)
. . . | . . . . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . 10. @source(l): ma-single-sends(x : A;
. . . | . . . . |                              k : B ⊕ rcv
. . . | . . . . |                              (l; tg) : T;
. . . | . . . . |                              k;
. . . | . . . . |                              l;
. . . | . . . . |                              [<tg,λs,v. f(s(x),v)>]
. . . | . . . . |                              )
. . . | . . . . |   ∈ Dsys
. . . | . . . . 11. ∀D:Dsys.
. . . | . . . . |   @source(l): ma-single-sends(x : A;
. . . | . . . . |                              k : B ⊕ rcv
. . . | . . . . |                              (l; tg) : T;
. . . | . . . . |                              k;
. . . | . . . . |                              l;
. . . | . . . . |                              [<tg,λs,v. f(s(x),v)>]
. . . | . . . . |                              ) ⊆ D
. . . | . . . . |   ⇒
. . . | . . . . |   D
. . . | . . . . |   realizes es.(∀x@0:Id.
. . . | . . . . |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . . |               & (∀e:E.
. . . | . . . . |                  loc(e) = source(l)
. . . | . . . . |                  ⇒
. . . | . . . . |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . . |                    )))
. . . | . . . . |               & (∀e:E.
. . . | . . . . |                  isrcv(e)
. . . | . . . . |                  ⇒
. . . | . . . . |                  lnk(e) = l
. . . | . . . . |                  ⇒
. . . | . . . . |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . . |                    )))
. . . | . . . . |               & (∀e:E.
. . . | . . . . |                  loc(e) = source(l)
. . . | . . . . |                  ⇒
. . . | . . . . |                  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);[<tg
. . . | . . . . |                                        ,λs,v.
. . . | . . . . |                                         f
. . . | . . . . |                                         (s(x)
. . . | . . . . |                                         ,v)>])))
. . . | . . . . 12. @source(l): ma-single-sends(x : A;
. . . | . . . . |                                k : B ⊕ rcv
. . . | . . . . |                                (l; tg) : T;
. . . | . . . . |                                k;
. . . | . . . . |                                l;
. . . | . . . . |                                [<tg,λs,v. f(s(x),v)>]
. . . | . . . . |                                )
. . . | . . . . |   ∈ Dsys
. . . | . . . . 13. D : Dsys
. . . | . . . . 14. @source(l): ma-single-sends(x : A;
. . . | . . . . |                                k : B ⊕ rcv
. . . | . . . . |                                (l; tg) : T;

. . . | . . . . |                                k;
. . . | . . . . |                                l;
. . . | . . . . |                                [<tg,λs,v. f(s(x),v)>]
. . . | . . . . |                                ) ⊆ D
. . . | . . . . 15. ∀D':Dsys.
. . . | . . . . |   D ⊆ D'
. . . | . . . . |   ⇒
. . . | . . . . |   (∀w:World, p:FairFifo.
. . . | . . . . |    PossibleWorld(D';w)
. . . | . . . . |    ⇒
. . . | . . . . |    (∀x@0:Id.
. . . | . . . . |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . |    & (∀e:E.
. . . | . . . . |       loc(e) = source(l)
. . . | . . . . |       ⇒
. . . | . . . . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |    & (∀e:E.
. . . | . . . . |       isrcv(e)
. . . | . . . . |       ⇒
. . . | . . . . |       lnk(e) = l
. . . | . . . . |       ⇒
. . . | . . . . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |    & (∀e:E.
. . . | . . . . |       loc(e) = source(l)
. . . | . . . . |       ⇒
. . . | . . . . |       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);[<tg
. . . | . . . . |                              ,λs,v. f(s(x),v)>])))
. . . | . . . . 16. D' : Dsys
. . . | . . . . 17. D ⊆ D'
. . . | . . . . 18. ∀w:World, p:FairFifo.
. . . | . . . . |   PossibleWorld(D';w)
. . . | . . . . |   ⇒
. . . | . . . . |   (∀x@0:Id.
. . . | . . . . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . |   & (∀e:E.
. . . | . . . . |      loc(e) = source(l)
. . . | . . . . |      ⇒
. . . | . . . . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |   & (∀e:E.
. . . | . . . . |      isrcv(e)
. . . | . . . . |      ⇒
. . . | . . . . |      lnk(e) = l
. . . | . . . . |      ⇒
. . . | . . . . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |   & (∀e:E.
. . . | . . . . |      loc(e) = source(l)
. . . | . . . . |      ⇒
. . . | . . . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . . . 19. w : World
. . . | . . . . 20. ∀p:FairFifo.
. . . | . . . . |   PossibleWorld(D';w)
. . . | . . . . |   ⇒
. . . | . . . . |   (∀x@0:Id.
. . . | . . . . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . |   & (∀e:E.
. . . | . . . . |      loc(e) = source(l)
. . . | . . . . |      ⇒
. . . | . . . . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |   & (∀e:E.
. . . | . . . . |      isrcv(e)
```

```
. . . | . . . . |      ⇒
. . . | . . . . |      lnk(e) = l
. . . | . . . . |      ⇒
. . . | . . . . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |   & (∀e:E.
. . . | . . . . |      loc(e) = source(l)
. . . | . . . . |      ⇒
. . . | . . . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . | . . . . 21. p : FairFifo
. . . | . . . . 22. PossibleWorld(D';w)
. . . | . . . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . 24. ∀e:E.
. . . | . . . . |   loc(e) = source(l)
. . . | . . . . |   ⇒
. . . | . . . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . |                             kind(e)))
. . . | . . . . 25. ∀e:E.
. . . | . . . . |   isrcv(e)
. . . | . . . . |   ⇒
. . . | . . . . |   lnk(e) = l
. . . | . . . . |   ⇒
. . . | . . . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . |                             kind(e)))
. . . | . . . . 26. ∀e:E.
. . . | . . . . |   loc(e) = source(l)
. . . | . . . . |   ⇒
. . . | . . . . |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . . . 27. vartype(source(l);x) ⊆r A
. . . | . . . . 28. ∀e:E.
. . . | . . . . |   loc(e) = source(l)
. . . | . . . . |   ⇒
. . . | . . . . |   kind(e) = k ⇒ (valtype(e) ⊆r B)
```

```
. . . | . . . . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . 30. e : E
. . . | . . . . 31. loc(e) = source(l)
. . . | . . . . 32. kind(e) = k
. . . | . . . . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . 34. ∀e':E.
. . . | . . . . |   (e' ∈ L)
. . . | . . . . |   ⟺
. . . | . . . . |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . |   =
. . . | . . . . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . 38. f((x when e),val(e)) ∈ T List
. . . | . . . . ⊢  map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . |  =
. . . | . . . . |  map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . | by StrongHypSubst -3 0
. . . | . . . . |
. . . | . . . . |\
. . . | . . . . | 1. x : Id
. . . | . . . . | 2. tg : Id
. . . | . . . . | 3. k : Knd
. . . | . . . . | 4. l : IdLnk
. . . | . . . . | 5. T : Type
. . . | . . . . | 6. A : Type
. . . | . . . . | 7. B : Type
. . . | . . . . | 8. f : A→B→(T List)
. . . | . . . . | 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . | 10. @source(l): ma-single-sends(x : A;
. . . | . . . . |                                 k : B ⊕ rcv
. . . | . . . . |                                 (l; tg) : T;
. . . | . . . . |                                 k;
. . . | . . . . |                                 l;
. . . | . . . . |                                 [<tg
. . . | . . . . |                                  ,λs,v. f(s(x),v)>])
. . . | . . . . |     ∈ Dsys
. . . | . . . . | 11. ∀D:Dsys.
. . . | . . . . |     @source(l): ma-single-sends(x : A;
. . . | . . . . |                                 k : B ⊕ rcv
. . . | . . . . |                                 (l; tg) : T;
. . . | . . . . |                                 k;
. . . | . . . . |                                 l;
. . . | . . . . |                                 [<tg
. . . | . . . . |                                  ,λs,v. f(s(x),v)>]
. . . | . . . . |                                 ) ⊆ D
. . . | . . . . |     ⇒
. . . | . . . . |     D
. . . | . . . . |     realizes es.(∀x@0:Id.
. . . | . . . . |                   vartype(source(l);x@0) ⊆r ...)
. . . | . . . . |                  & (∀e:E.
. . . | . . . . |                     loc(e) = source(l)
. . . | . . . . |                     ⇒
. . . | . . . . |                     (valtype(e) ⊆r ma-valtype
```

```
. . . | . . . . |                    (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . . |                     )))

. . . | . . . . |                & (∀e:E.
. . . | . . . . |                   isrcv(e)
. . . | . . . . |                   ⇒
. . . | . . . . |                   lnk(e) = l
. . . | . . . . |                   ⇒
. . . | . . . . |                   (valtype(e) ⊆r ma-valtype
. . . | . . . . |                    (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . . |                     )))
. . . | . . . . |                & (∀e:E.
. . . | . . . . |                   loc(e) = source(l)
. . . | . . . . |                   ⇒
. . . | . . . . |                   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);[<tg
. . . | . . . . |                                        ,λs,v.
. . . | . . . . |                                         f
. . . | . . . . |                                         (s(x)
. . . | . . . . |                                         ,v)>])))
. . . | . . . . | 12. @source(l): ma-single-sends(x : A;
. . . | . . . . |                                 k : B ⊕ rcv
. . . | . . . . |                                 (l; tg) : T;
. . . | . . . . |                                 k;
. . . | . . . . |                                 l;
. . . | . . . . |                                 [<tg
. . . | . . . . |                                  ,λs,v. f(s(x),v)>])
. . . | . . . . |     ∈ Dsys
. . . | . . . . | 13. D : Dsys
. . . | . . . . | 14. @source(l): ma-single-sends(x : A;
. . . | . . . . |                                 k : B ⊕ rcv
. . . | . . . . |                                 (l; tg) : T;
. . . | . . . . |                                 k;
. . . | . . . . |                                 l;
. . . | . . . . |                                 [<tg
. . . | . . . . |                                  ,λs,v. f(s(x),v)>]
. . . | . . . . |                                 ) ⊆ D
. . . | . . . . | 15. ∀D':Dsys.
. . . | . . . . |     D ⊆ D'
```

```
. . . | . . . . |     ⇒
. . . | . . . . |     (∀w:World, p:FairFifo.
. . . | . . . . |      PossibleWorld(D';w)
. . . | . . . . |      ⇒
. . . | . . . . |      (∀x@0:Id.
. . . | . . . . |       vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . |      & (∀e:E.
. . . | . . . . |         loc(e) = source(l)
. . . | . . . . |         ⇒
. . . | . . . . |         (valtype(e) ⊆r ma-valtype
. . . | . . . . |          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |      & (∀e:E.
. . . | . . . . |         isrcv(e)
. . . | . . . . |         ⇒
. . . | . . . . |         lnk(e) = l
. . . | . . . . |         ⇒
. . . | . . . . |         (valtype(e) ⊆r ma-valtype
. . . | . . . . |          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |      & (∀e:E.
. . . | . . . . |         loc(e) = source(l)
. . . | . . . . |         ⇒
. . . | . . . . |         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);[<tg
. . . | . . . . |                                ,λs,v. f(s(x),v)>])))))
. . . | . . . . | 16. D' : Dsys
. . . | . . . . | 17. D ⊆ D'
. . . | . . . . | 18. ∀w:World, p:FairFifo.
. . . | . . . . |     PossibleWorld(D';w)
. . . | . . . . |     ⇒
. . . | . . . . |     (∀x@0:Id.
. . . | . . . . |      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . |     & (∀e:E.
. . . | . . . . |        loc(e) = source(l)
. . . | . . . . |        ⇒
. . . | . . . . |        (valtype(e) ⊆r ma-valtype
. . . | . . . . |         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |     & (∀e:E.
. . . | . . . . |        isrcv(e)
. . . | . . . . |        ⇒
. . . | . . . . |        lnk(e) = l
. . . | . . . . |        ⇒
. . . | . . . . |        (valtype(e) ⊆r ma-valtype
. . . | . . . . |         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |     & (∀e:E.
```

```
. . . | . . . . |        loc(e) = source(l)
. . . | . . . . |        ⇒
. . . | . . . . |        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);[<tg
. . . | . . . . |                               ,λs,v. f(s(x),v)>])))
. . . | . . . . | 19. w : World
. . . | . . . . | 20. ∀p:FairFifo.
. . . | . . . . |     PossibleWorld(D';w)
. . . | . . . . |     ⇒
. . . | . . . . |     (∀x@0:Id.
. . . | . . . . |      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . |     & (∀e:E.
. . . | . . . . |        loc(e) = source(l)
. . . | . . . . |        ⇒
. . . | . . . . |        (valtype(e) ⊆r ma-valtype
. . . | . . . . |         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |     & (∀e:E.
. . . | . . . . |        isrcv(e)
. . . | . . . . |        ⇒
. . . | . . . . |        lnk(e) = l
. . . | . . . . |        ⇒
. . . | . . . . |        (valtype(e) ⊆r ma-valtype
. . . | . . . . |         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . |     & (∀e:E.
. . . | . . . . |        loc(e) = source(l)
. . . | . . . . |        ⇒
. . . | . . . . |        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);[<tg
. . . | . . . . |                               ,λs,v. f(s(x),v)>])))
. . . | . . . . | 21. p : FairFifo
. . . | . . . . | 22. PossibleWorld(D';w)
. . . | . . . . | 23. ∀x@0:Id.
. . . | . . . . |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top
```

```
. . . | . . . . | 24. ∀e:E.
. . . | . . . . |     loc(e) = source(l)
. . . | . . . . |     ⇒
. . . | . . . . |     (valtype(e) ⊆r ma-valtype
. . . | . . . . |      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . | 25. ∀e:E.
. . . | . . . . |     isrcv(e)
. . . | . . . . |     ⇒
. . . | . . . . |     lnk(e) = l
. . . | . . . . |     ⇒
. . . | . . . . |     (valtype(e) ⊆r ma-valtype
. . . | . . . . |      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . | 26. ∀e:E.
. . . | . . . . |     loc(e) = source(l)
. . . | . . . . |     ⇒
. . . | . . . . |     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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . . . | 27. vartype(source(l);x) ⊆r A
. . . | . . . . | 28. ∀e:E.
. . . | . . . . |     loc(e) = source(l)
. . . | . . . . |     ⇒
. . . | . . . . |     kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . | 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . | 30. e : E
. . . | . . . . | 31. loc(e) = source(l)
. . . | . . . . | 32. kind(e) = k
. . . | . . . . | 33. L : {e':E| isrcv(e') & lnk(e') = l } List

. . . | . . . . | 34. ∀e':E.
. . . | . . . . |     (e' ∈ L)
. . . | . . . . |     ⟺
. . . | . . . . |     isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . |     =
. . . | . . . . |     map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . | 38. f((x when e),val(e)) ∈ T List
. . . | . . . . | ⊢  map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . |    =
. . . | . . . . |    map(λx@0.<tg,x@0>;f((x when e),val(e)))  by Auto
. . . | . . . . |
. . . | . . . . .\
. . . | . . . . . 1. x : Id
. . . | . . . . . 2. tg : Id
. . . | . . . . . 3. k : Knd
```

```
. . . | . . . . . 4. l : IdLnk
. . . | . . . . . 5. T : Type
. . . | . . . . . 6. A : Type
. . . | . . . . . 7. B : Type
. . . | . . . . . 8. f : A→B→(T List)
. . . | . . . . . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . 10. @source(l): ma-single-sends(x : A;
. . . | . . . . . |                              k : B ⊕ rcv
. . . | . . . . . |                              (l; tg) : T;
. . . | . . . . . |                              k;
. . . | . . . . . |                              l;
. . . | . . . . . |                              [<tg
. . . | . . . . . |                               ,λs,v. f(s(x),v)>])
. . . | . . . . . |   ∈ Dsys
. . . | . . . . . 11. ∀D:Dsys.
. . . | . . . . . |   @source(l): ma-single-sends(x : A;
. . . | . . . . . |                              k : B ⊕ rcv
. . . | . . . . . |                              (l; tg) : T;
. . . | . . . . . |                              k;
. . . | . . . . . |                              l;
. . . | . . . . . |                              [<tg
. . . | . . . . . |                               ,λs,v. f(s(x),v)>]
. . . | . . . . . |                              ) ⊆ D
. . . | . . . . . |   ⇒
. . . | . . . . . |   D
. . . | . . . . . |   realizes es.(∀x@0:Id.
. . . | . . . . . |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . |               & (∀e:E.
. . . | . . . . . |                  loc(e) = source(l)
. . . | . . . . . |                  ⇒
. . . | . . . . . |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . . . |                    )))
. . . | . . . . . |               & (∀e:E.
. . . | . . . . . |                  isrcv(e)
. . . | . . . . . |                  ⇒
. . . | . . . . . |                  lnk(e) = l
. . . | . . . . . |                  ⇒
. . . | . . . . . |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e)
. . . | . . . . . |                    )))
. . . | . . . . . |               & (∀e:E.
. . . | . . . . . |                  loc(e) = source(l)
. . . | . . . . . |                  ⇒
. . . | . . . . . |                  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);[<tg
. . . | . . . . . |                                       ,λs,v.
. . . | . . . . . |                                        f
. . . | . . . . . |                                        (s(x)
. . . | . . . . . |                                        ,v)>])))
. . . | . . . . . 12. @source(l): ma-single-sends(x : A;
. . . | . . . . . |                                k : B ⊕ rcv
. . . | . . . . . |                                (l; tg) : T;
. . . | . . . . . |                                k;
. . . | . . . . . |                                l;
. . . | . . . . . |                                [<tg
. . . | . . . . . |                                 ,λs,v. f(s(x),v)>])
. . . | . . . . . |   ∈ Dsys
. . . | . . . . . 13. D : Dsys
. . . | . . . . . 14. @source(l): ma-single-sends(x : A;
. . . | . . . . . |                                k : B ⊕ rcv
. . . | . . . . . |                                (l; tg) : T;
. . . | . . . . . |                                k;
. . . | . . . . . |                                l;
. . . | . . . . . |                                [<tg
. . . | . . . . . |                                 ,λs,v. f(s(x),v)>]
. . . | . . . . . |                                ) ⊆ D
. . . | . . . . . 15. ∀D':Dsys.
. . . | . . . . . |   D ⊆ D'
. . . | . . . . . |   ⇒
. . . | . . . . . |   (∀w:World, p:FairFifo.
. . . | . . . . . |    PossibleWorld(D';w)
. . . | . . . . . |    ⇒
. . . | . . . . . |    (∀x@0:Id.
. . . | . . . . . |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . |    & (∀e:E.
. . . | . . . . . |       loc(e) = source(l)
. . . | . . . . . |       ⇒
. . . | . . . . . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . |    & (∀e:E.
. . . | . . . . . |       isrcv(e)
. . . | . . . . . |       ⇒
. . . | . . . . . |       lnk(e) = l
. . . | . . . . . |       ⇒
. . . | . . . . . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . |    & (∀e:E.
. . . | . . . . . |       loc(e) = source(l)
. . . | . . . . . |       ⇒
. . . | . . . . . |       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);[<tg
. . . | . . . . . |                              ,λs,v. f(s(x),v)>])))
. . . | . . . . . 16. D' : Dsys
. . . | . . . . . 17. D ⊆ D'
. . . | . . . . . 18. ∀w:World, p:FairFifo.
. . . | . . . . . |   PossibleWorld(D';w)
. . . | . . . . . |   ⇒
. . . | . . . . . |   (∀x@0:Id.
. . . | . . . . . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . |   & (∀e:E.
. . . | . . . . . |      loc(e) = source(l)
. . . | . . . . . |      ⇒
. . . | . . . . . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . |   & (∀e:E.
. . . | . . . . . |      isrcv(e)
. . . | . . . . . |      ⇒
. . . | . . . . . |      lnk(e) = l
. . . | . . . . . |      ⇒
. . . | . . . . . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . |   & (∀e:E.
. . . | . . . . . |      loc(e) = source(l)
. . . | . . . . . |      ⇒
. . . | . . . . . |      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);[<tg
. . . | . . . . . |                              ,λs,v. f(s(x),v)>])))
. . . | . . . . . 19. w : World
. . . | . . . . . 20. ∀p:FairFifo.
. . . | . . . . . |   PossibleWorld(D';w)
. . . | . . . . . |   ⇒
. . . | . . . . . |   (∀x@0:Id.
. . . | . . . . . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . |   & (∀e:E.
. . . | . . . . . |      loc(e) = source(l)
```

```
. . . | . . . . . |      ⇒
. . . | . . . . . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . |   & (∀e:E.
. . . | . . . . . |      isrcv(e)
. . . | . . . . . |      ⇒
. . . | . . . . . |      lnk(e) = l
. . . | . . . . . |      ⇒
. . . | . . . . . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . |   & (∀e:E.
. . . | . . . . . |      loc(e) = source(l)
. . . | . . . . . |      ⇒
. . . | . . . . . |      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);[<tg
. . . | . . . . . |                              ,λs,v. f(s(x),v)>])))
. . . | . . . . . 21. p : FairFifo
. . . | . . . . . 22. PossibleWorld(D';w)
. . . | . . . . . 23. ∀x@0:Id.
. . . | . . . . . |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . 24. ∀e:E.
. . . | . . . . . |   loc(e) = source(l)
. . . | . . . . . |   ⇒
. . . | . . . . . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . 25. ∀e:E.
. . . | . . . . . |   isrcv(e)
. . . | . . . . . |   ⇒
. . . | . . . . . |   lnk(e) = l
. . . | . . . . . |   ⇒
. . . | . . . . . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . 26. ∀e:E.
. . . | . . . . . |   loc(e) = source(l)
. . . | . . . . . |   ⇒
. . . | . . . . . |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . . . . 27. vartype(source(l);x) ⊆r A
. . . | . . . . . 28. ∀e:E.
. . . | . . . . . |   loc(e) = source(l)
. . . | . . . . . |   ⇒
. . . | . . . . . |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . 30. e : E
. . . | . . . . . 31. loc(e) = source(l)
. . . | . . . . . 32. kind(e) = k
. . . | . . . . . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . 34. ∀e':E.
. . . | . . . . . |   (e' ∈ L)
. . . | . . . . . |   ⟺
. . . | . . . . . |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . |   =
. . . | . . . . . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . 39. z :
. . . | . . . . . |   (tg@0:Id
. . . | . . . . . |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . |                rcv(l; tg@0))) List
. . . | . . . . . 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . ⊢  (z = map(λx@0.<tg,x@0>;f((x when e),val(e))))
. . . | . . . . . |  ∈ Prop  by Assert (z ∈ (Id×T) List)
. . . | . . . . . |
. . . | . . . . . |\
. . . | . . . . . | .....assertion .....
. . . | . . . . . | 1. x : Id
. . . | . . . . . | 2. tg : Id
. . . | . . . . . | 3. k : Knd
. . . | . . . . . | 4. l : IdLnk
. . . | . . . . . | 5. T : Type
. . . | . . . . . | 6. A : Type
. . . | . . . . . | 7. B : Type
. . . | . . . . . | 8. f : A→B→(T List)
. . . | . . . . . | 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | 10. @source(l): ma-single-sends(x : A;
. . . | . . . . . | | |                              k : B ⊕ rcv
. . . | . . . . . | | |                              (l; tg) : T;
. . . | . . . . . | | |                              k;
. . . | . . . . . | | |                              l;
. . . | . . . . . | | |                              [<tg
. . . | . . . . . | | |                               ,λs,v. f
. . . | . . . . . | | |                                      (s(x)
. . . | . . . . . | | |                                      ,v)>])
. . . | . . . . . | | |   ∈ Dsys
. . . | . . . . . | 11. ∀D:Dsys.
. . . | . . . . . | | |   @source(l): ma-single-sends(x : A;
. . . | . . . . . | | |                              k : B ⊕ rcv
```

```
. . . | . . . . . | |                                (l; tg) : T;
. . . | . . . . . | |                                k;
. . . | . . . . . | |                                l;
. . . | . . . . . | |                                [<tg
. . . | . . . . . | |                                 ,λs,v. f
. . . | . . . . . | |                                        (s(x)
. . . | . . . . . | |                                        ,v)>]) ⊆ D

. . . | . . . . . | |   ⇒
. . . | . . . . . | |   D
. . . | . . . . . | |   realizes es.(∀x@0:Id.
. . . | . . . . . | |                 vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . | |               & (∀e:E.
. . . | . . . . . | |                    loc(e) = source(l)
. . . | . . . . . | |                    ⇒
. . . | . . . . . | |                    (valtype(e) ⊆r ma-valtype
. . . | . . . . . | |                     (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | |                      kind(e))))
. . . | . . . . . | |               & (∀e:E.
. . . | . . . . . | |                    isrcv(e)
. . . | . . . . . | |                    ⇒
. . . | . . . . . | |                    lnk(e) = l
. . . | . . . . . | |                    ⇒
. . . | . . . . . | |                    (valtype(e) ⊆r ma-valtype
. . . | . . . . . | |                     (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | |                      kind(e))))
. . . | . . . . . | |               & (∀e:E.
. . . | . . . . . | |                    loc(e) = source(l)
. . . | . . . . . | |                    ⇒
. . . | . . . . . | |                    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);[<tg
. . . | . . . . . | |                                          ,λs,v.
. . . | . . . . . | |                                           f
. . . | . . . . . | |                                           (s(x)
. . . | . . . . . | |                                           ,v)>])))
. . . | . . . . . | 12. @source(l): ma-single-sends(x : A;
. . . | . . . . . | |                                k : B ⊕ rcv
. . . | . . . . . | |                                (l; tg) : T;
```

```
. . . | . . . . . | |                                  k;
. . . | . . . . . | |                                  l;
. . . | . . . . . | |                                  [<tg
. . . | . . . . . | |                                   ,λs,v. f
. . . | . . . . . | |                                          (s(x)
. . . | . . . . . | |                                          ,v)>])
. . . | . . . . . | |   ∈ Dsys
. . . | . . . . . | 13. D : Dsys
. . . | . . . . . | 14. @source(l): ma-single-sends(x : A;
. . . | . . . . . | |                                  k : B ⊕ rcv
. . . | . . . . . | |                                  (l; tg) : T;
. . . | . . . . . | |                                  k;
. . . | . . . . . | |                                  l;
. . . | . . . . . | |                                  [<tg
. . . | . . . . . | |                                   ,λs,v. f
. . . | . . . . . | |                                          (s(x)
. . . | . . . . . | |                                          ,v)>]) ⊆ D
. . . | . . . . . | 15. ∀D':Dsys.
. . . | . . . . . | |   D ⊆ D'
. . . | . . . . . | |   ⇒
. . . | . . . . . | |   (∀w:World, p:FairFifo.
. . . | . . . . . | |    PossibleWorld(D';w)
. . . | . . . . . | |    ⇒
. . . | . . . . . | |    (∀x@0:Id.
. . . | . . . . . | |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | |    & (∀e:E.
. . . | . . . . . | |       loc(e) = source(l)
. . . | . . . . . | |       ⇒
. . . | . . . . . | |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | |    & (∀e:E.
. . . | . . . . . | |       isrcv(e)
. . . | . . . . . | |       ⇒
. . . | . . . . . | |       lnk(e) = l
. . . | . . . . . | |       ⇒
. . . | . . . . . | |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | |    & (∀e:E.
. . . | . . . . . | |       loc(e) = source(l)
. . . | . . . . . | |       ⇒
. . . | . . . . . | |       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);[<tg
```

```
. . . | . . . . . | |                              ,λs,v. f
. . . | . . . . . | |                                     (s(x)
. . . | . . . . . | |                                     ,v)>])))))
. . . | . . . . . | 16. D' : Dsys
. . . | . . . . . | 17. D ⊆ D'
. . . | . . . . . | 18. ∀w:World, p:FairFifo.
. . . | . . . . . | |   PossibleWorld(D';w)
. . . | . . . . . | |   ⇒
. . . | . . . . . | |   (∀x@0:Id.
. . . | . . . . . | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | |   & (∀e:E.
. . . | . . . . . | |      loc(e) = source(l)
. . . | . . . . . | |      ⇒
. . . | . . . . . | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | |   & (∀e:E.
. . . | . . . . . | |      isrcv(e)
. . . | . . . . . | |      ⇒
. . . | . . . . . | |      lnk(e) = l
. . . | . . . . . | |      ⇒
. . . | . . . . . | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | |   & (∀e:E.
. . . | . . . . . | |      loc(e) = source(l)
. . . | . . . . . | |      ⇒
. . . | . . . . . | |      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);[<tg
. . . | . . . . . | |                              ,λs,v. f(s(x),v)>])))
. . . | . . . . . | 19. w : World
. . . | . . . . . | 20. ∀p:FairFifo.
. . . | . . . . . | |   PossibleWorld(D';w)
. . . | . . . . . | |   ⇒
. . . | . . . . . | |   (∀x@0:Id.
. . . | . . . . . | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | |   & (∀e:E.
. . . | . . . . . | |      loc(e) = source(l)
. . . | . . . . . | |      ⇒
. . . | . . . . . | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | |   & (∀e:E.
. . . | . . . . . | |      isrcv(e)
. . . | . . . . . | |      ⇒
. . . | . . . . . | |      lnk(e) = l
. . . | . . . . . | |      ⇒
```

```
. . . | . . . . . | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | |   & (∀e:E.
. . . | . . . . . | |      loc(e) = source(l)
. . . | . . . . . | |      ⇒
. . . | . . . . . | |      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);[<tg
. . . | . . . . . | |                            ,λs,v. f(s(x),v)>]))))
. . . | . . . . . | 21. p : FairFifo
. . . | . . . . . | 22. PossibleWorld(D';w)
. . . | . . . . . | 23. ∀x@0:Id.
. . . | . . . . . | |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | 24. ∀e:E.
. . . | . . . . . | |   loc(e) = source(l)
. . . | . . . . . | |   ⇒
. . . | . . . . . | |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | 25. ∀e:E.
. . . | . . . . . | |   isrcv(e)
. . . | . . . . . | |   ⇒
. . . | . . . . . | |   lnk(e) = l
. . . | . . . . . | |   ⇒
. . . | . . . . . | |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | 26. ∀e:E.
. . . | . . . . . | |   loc(e) = source(l)
. . . | . . . . . | |   ⇒
. . . | . . . . . | |   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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . . . . | 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | 28. ∀e:E.
. . . | . . . . . | |   loc(e) = source(l)
. . . | . . . . . | |   ⇒
```

```
. . . | . . . . . | |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | 29. ∀e:E.
. . . | . . . . . | |   kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | 30. e : E
. . . | . . . . . | 31. loc(e) = source(l)
. . . | . . . . . | 32. kind(e) = k
. . . | . . . . . | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | 34. ∀e':E.
. . . | . . . . . | |   (e' ∈ L)
. . . | . . . . . | |   ⟺
. . . | . . . . . | |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | |   =
. . . | . . . . . | |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | 39. z :
. . . | . . . . . | |   (tg@0:Id
. . . | . . . . . | |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | |                rcv(l; tg@0))) List
. . . | . . . . . | 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | ⊢  z ∈ (Id×T) List
. . . | . . . . . | | by Assert
. . . | . . . . . | |    (z
. . . | . . . . . | |     ∈ {p:(tg@0:Id
. . . | . . . . . | |           ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | |                        rcv(l; tg@0)))
. . . | . . . . . | |        | 1of(p) = tg } List)
. . . | . . . . . | |
. . . | . . . . . | |\
. . . | . . . . . | | .....assertion .....
. . . | . . . . . | | 1. x : Id
. . . | . . . . . | | 2. tg : Id
. . . | . . . . . | | 3. k : Knd
. . . | . . . . . | | 4. l : IdLnk
. . . | . . . . . | | 5. T : Type
. . . | . . . . . | | 6. A : Type
. . . | . . . . . | | 7. B : Type
. . . | . . . . . | | 8. f : A→B→(T List)
. . . | . . . . . | | 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | | 10. @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | |   ∈ Dsys
. . . | . . . . . | | 11. ∀D:Dsys.
. . . | . . . . . | | |   @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
```

```
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   D
. . . | . . . . . | | |   realizes es.(∀x@0:Id.
. . . | . . . . . | | |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . | | |               & (∀e:E.
. . . | . . . . . | | |                  loc(e) = source(l)
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |                   (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                    kind(e))))
. . . | . . . . . | | |               & (∀e:E.
. . . | . . . . . | | |                  isrcv(e)
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  lnk(e) = l
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |                   (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                    kind(e))))
. . . | . . . . . | | |               & (∀e:E.
. . . | . . . . . | | |                  loc(e) = source(l)
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  kind(e) = k
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  ...)
. . . | . . . . . | | 12. @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | |   ∈ Dsys
. . . | . . . . . | | 13. D : Dsys
. . . | . . . . . | | 14. @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | 15. ∀D':Dsys.
. . . | . . . . . | | |   D ⊆ D'
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (∀w:World, p:FairFifo.
. . . | . . . . . | | |    PossibleWorld(D';w)
. . . | . . . . . | | |    ⇒
. . . | . . . . . | | |    (∀x@0:Id.
. . . | . . . . . | | |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |    & (∀e:E.
. . . | . . . . . | | |       loc(e) = source(l)
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |    & (∀e:E.
. . . | . . . . . | | |       isrcv(e)
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       lnk(e) = l
```

```
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |    & (∀e:E.
. . . | . . . . . | | |       loc(e) = source(l)
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       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);[<tg
. . . | . . . . . | | |                              ,λs,v. f
. . . | . . . . . | | |                                     (s(x)
. . . | . . . . . | | |                                     ,v)>])))))
. . . | . . . . . | | 16. D' : Dsys
. . . | . . . . . | | 17. D ⊆ D'
. . . | . . . . . | | 18. ∀w:World, p:FairFifo.
. . . | . . . . . | | |   PossibleWorld(D';w)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (∀x@0:Id.
. . . | . . . . . | | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      isrcv(e)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      lnk(e) = l
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      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);[<tg
. . . | . . . . . | | |                              ,λs,v. f
. . . | . . . . . | | |                                     (s(x)
. . . | . . . . . | | |                                     ,v)>])))
. . . | . . . . . | | 19. w : World
. . . | . . . . . | | 20. ∀p:FairFifo.
. . . | . . . . . | | |   PossibleWorld(D';w)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (∀x@0:Id.
. . . | . . . . . | | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      isrcv(e)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      lnk(e) = l
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      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);[<tg
. . . | . . . . . | | |                              ,λs,v. f
. . . | . . . . . | | |                                     (s(x)
. . . | . . . . . | | |                                     ,v)>])))
. . . | . . . . . | | 21. p : FairFifo
. . . | . . . . . | | 22. PossibleWorld(D';w)
. . . | . . . . . | | 23. ∀x@0:Id.
. . . | . . . . . | | |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
```

```
. . . | . . . . . | | 24. ∀e:E.
. . . | . . . . . | | |   loc(e) = source(l)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | 25. ∀e:E.
. . . | . . . . . | | |   isrcv(e)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   lnk(e) = l
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | 26. ∀e:E.
. . . | . . . . . | | |   loc(e) = source(l)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   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);[<tg
. . . | . . . . . | | |                          ,λs,v. f(s(x),v)>]))
. . . | . . . . . | | 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | | 28. ∀e:E.
. . . | . . . . . | | |   loc(e) = source(l)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | | 29. ∀e:E.
. . . | . . . . . | | |   kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | | 30. e : E
. . . | . . . . . | | 31. loc(e) = source(l)
. . . | . . . . . | | 32. kind(e) = k
. . . | . . . . . | | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | | 34. ∀e':E.
. . . | . . . . . | | |   (e' ∈ L)
. . . | . . . . . | | |   ⟺
. . . | . . . . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | | |   =
. . . | . . . . . | | |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | | 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | | 39. z :
. . . | . . . . . | | |   (tg@0:Id
. . . | . . . . . | | |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                rcv(l; tg@0))) List
. . . | . . . . . | | 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
```

```
. . . | . . . . . | | ⊢ z
. . . | . . . . . | | |  ∈ {p:(tg@0:Id
. . . | . . . . . | | |        ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                    rcv(l; tg@0)))
. . . | . . . . . | | |     | 1of(p) = tg } List
. . . | . . . . . | | | by BackThru
. . . | . . . . . | | |    Thm* ∀T:Type, L:T List, P:(T→Prop).
. . . | . . . . . | | |         (∀x∈L.P(x)) ⇒ L ∈ {x:T| P(x) } List
. . . | . . . . . | | |    {list_DASH_set_DASH_type2}
. . . | . . . . . | | |
. . . | . . . . . | | 1. x : Id
. . . | . . . . . | | 2. tg : Id
. . . | . . . . . | | 3. k : Knd
. . . | . . . . . | | 4. l : IdLnk
. . . | . . . . . | | 5. T : Type
. . . | . . . . . | | 6. A : Type
. . . | . . . . . | | 7. B : Type
. . . | . . . . . | | 8. f : A→B→(T List)
. . . | . . . . . | | 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | | 10. @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | |   ∈ Dsys
. . . | . . . . . | | 11. ∀D:Dsys.
. . . | . . . . . | | |   @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   D
. . . | . . . . . | | |   realizes es.(∀x@0:Id.
. . . | . . . . . | | |                 vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . | | |               & (∀e:E.
. . . | . . . . . | | |                  loc(e) = source(l)
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |                   (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                    kind(e))))
. . . | . . . . . | | |               & (∀e:E.
. . . | . . . . . | | |                  isrcv(e)
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  lnk(e) = l
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |                   (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                    kind(e))))
. . . | . . . . . | | |               & (∀e:E.
. . . | . . . . . | | |                  loc(e) = source(l)
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  kind(e) = k
```

```
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  ...)
. . . | . . . . . | | 12. @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | |   ∈ Dsys
. . . | . . . . . | | 13. D : Dsys
. . . | . . . . . | | 14. @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | 15. ∀D':Dsys.
. . . | . . . . . | | |   D ⊆ D'
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (∀w:World, p:FairFifo.
. . . | . . . . . | | |    PossibleWorld(D';w)
. . . | . . . . . | | |    ⇒
. . . | . . . . . | | |    (∀x@0:Id.
. . . | . . . . . | | |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |    & (∀e:E.
. . . | . . . . . | | |       loc(e) = source(l)
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |    & (∀e:E.
. . . | . . . . . | | |       isrcv(e)
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       lnk(e) = l
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |    & (∀e:E.
. . . | . . . . . | | |       loc(e) = source(l)
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       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);[<tg
```

```
. . . | . . . . . | | |                              ,λs,v. f
. . . | . . . . . | | |                                     (s(x)
. . . | . . . . . | | |                                     ,v)>])))))
. . . | . . . . . | | 16. D' : Dsys
. . . | . . . . . | | 17. D ⊆ D'
. . . | . . . . . | | 18. ∀w:World, p:FairFifo.
. . . | . . . . . | | |   PossibleWorld(D';w)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (∀x@0:Id.
. . . | . . . . . | | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      isrcv(e)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      lnk(e) = l
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      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);[<tg
. . . | . . . . . | | |                              ,λs,v. f
. . . | . . . . . | | |                                     (s(x)
. . . | . . . . . | | |                                     ,v)>])))
. . . | . . . . . | | 19. w : World
. . . | . . . . . | | 20. ∀p:FairFifo.
. . . | . . . . . | | |   PossibleWorld(D';w)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (∀x@0:Id.
. . . | . . . . . | | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
```

```
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      isrcv(e)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      lnk(e) = l
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      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);[<tg
. . . | . . . . . | | |                              ,λs,v. f
. . . | . . . . . | | |                                     (s(x)
. . . | . . . . . | | |                                     ,v)>])))
. . . | . . . . . | | 21. p : FairFifo
. . . | . . . . . | | 22. PossibleWorld(D';w)
. . . | . . . . . | | 23. ∀x@0:Id.
. . . | . . . . . | | |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | | 24. ∀e:E.
. . . | . . . . . | | |   loc(e) = source(l)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | 25. ∀e:E.
. . . | . . . . . | | |   isrcv(e)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   lnk(e) = l
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | 26. ∀e:E.
. . . | . . . . . | | |   loc(e) = source(l)

. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   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);[<tg
. . . | . . . . . | | |                          ,λs,v. f(s(x),v)>]))
. . . | . . . . . | | 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | | 28. ∀e:E.
. . . | . . . . . | | |   loc(e) = source(l)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | | 29. ∀e:E.
. . . | . . . . . | | |   kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | | 30. e : E
. . . | . . . . . | | 31. loc(e) = source(l)
. . . | . . . . . | | 32. kind(e) = k
. . . | . . . . . | | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | | 34. ∀e':E.
. . . | . . . . . | | |   (e' ∈ L)
. . . | . . . . . | | |   ⟺
. . . | . . . . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | | |   =
. . . | . . . . . | | |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | | 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | | 39. z :
. . . | . . . . . | | |   (tg@0:Id
. . . | . . . . . | | |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                rcv(l; tg@0))) List
. . . | . . . . . | | 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | ⊢  (∀p∈z.1of(p) = tg)  by Unfold ‘l_all‘ 0
. . . | . . . . . | | |
. . . | . . . . . | | 1. x : Id
. . . | . . . . . | | 2. tg : Id
. . . | . . . . . | | 3. k : Knd
. . . | . . . . . | | 4. l : IdLnk
. . . | . . . . . | | 5. T : Type
. . . | . . . . . | | 6. A : Type
. . . | . . . . . | | 7. B : Type
. . . | . . . . . | | 8. f : A→B→(T List)
. . . | . . . . . | | 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | | 10. @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | |   ∈ Dsys
. . . | . . . . . | | 11. ∀D:Dsys.
. . . | . . . . . | | |   @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
```

```
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   D
. . . | . . . . . | | |   realizes es.(∀x@0:Id.
. . . | . . . . . | | |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . | | |               & (∀e:E.
. . . | . . . . . | | |                  loc(e) = source(l)
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |                   (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                    kind(e))))
. . . | . . . . . | | |               & (∀e:E.
. . . | . . . . . | | |                  isrcv(e)
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  lnk(e) = l
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |                   (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                    kind(e))))
. . . | . . . . . | | |               & (∀e:E.
. . . | . . . . . | | |                  loc(e) = source(l)
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  kind(e) = k
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  ...)
. . . | . . . . . | | 12. @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | |   ∈ Dsys
. . . | . . . . . | | 13. D : Dsys
. . . | . . . . . | | 14. @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | 15. ∀D':Dsys.
. . . | . . . . . | | |   D ⊆ D'
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (∀w:World, p:FairFifo.
. . . | . . . . . | | |    PossibleWorld(D';w)
. . . | . . . . . | | |    ⇒
. . . | . . . . . | | |    (∀x@0:Id.
. . . | . . . . . | | |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |    & (∀e:E.
. . . | . . . . . | | |       loc(e) = source(l)
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |    & (∀e:E.
```

```
. . . | . . . . . | | |       isrcv(e)
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       lnk(e) = l
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |    & (∀e:E.
. . . | . . . . . | | |       loc(e) = source(l)
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       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);[<tg
. . . | . . . . . | | |                              ,λs,v. f
. . . | . . . . . | | |                                     (s(x)
. . . | . . . . . | | |                                     ,v)>])))))
. . . | . . . . . | | 16. D' : Dsys
. . . | . . . . . | | 17. D ⊆ D'
. . . | . . . . . | | 18. ∀w:World, p:FairFifo.
. . . | . . . . . | | |   PossibleWorld(D';w)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (∀x@0:Id.
. . . | . . . . . | | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      isrcv(e)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      lnk(e) = l
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      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);[<tg
. . . | . . . . . | | |                            ,λs,v. f
. . . | . . . . . | | |                                   (s(x)
. . . | . . . . . | | |                                   ,v)>])))
. . . | . . . . . | | 19. w : World
. . . | . . . . . | | 20. ∀p:FairFifo.
. . . | . . . . . | | |   PossibleWorld(D';w)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (∀x@0:Id.
. . . | . . . . . | | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      isrcv(e)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      lnk(e) = l
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      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);[<tg
. . . | . . . . . | | |                            ,λs,v. f
. . . | . . . . . | | |                                   (s(x)
. . . | . . . . . | | |                                   ,v)>])))
. . . | . . . . . | | 21. p : FairFifo
```

```
. . . | . . . . . | | 22. PossibleWorld(D';w)
. . . | . . . . . | | 23. ∀x@0:Id.
. . . | . . . . . | | |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | | 24. ∀e:E.
. . . | . . . . . | | |   loc(e) = source(l)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | 25. ∀e:E.
. . . | . . . . . | | |   isrcv(e)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   lnk(e) = l
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | 26. ∀e:E.
. . . | . . . . . | | |   loc(e) = source(l)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   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);[<tg
. . . | . . . . . | | |                         ,λs,v. f(s(x),v)>]))
. . . | . . . . . | | 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | | 28. ∀e:E.
. . . | . . . . . | | |   loc(e) = source(l)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | | 29. ∀e:E.
. . . | . . . . . | | |   kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | | 30. e : E
. . . | . . . . . | | 31. loc(e) = source(l)
. . . | . . . . . | | 32. kind(e) = k
. . . | . . . . . | | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | | 34. ∀e':E.
. . . | . . . . . | | |   (e' ∈ L)
. . . | . . . . . | | |   ⟺
. . . | . . . . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | | |   =
. . . | . . . . . | | |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | | 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | | 39. z :
. . . | . . . . . | | |   (tg@0:Id
```

```
. . . | . . . . . | | |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                rcv(l; tg@0))) List
. . . | . . . . . | | 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | 41. p1 :
. . . | . . . . . | | |   tg@0:Id
. . . | . . . . . | | |   ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |               rcv(l; tg@0))
. . . | . . . . . | | 42. (p1 ∈ z)
. . . | . . . . . | | ⊢  1of(p1) = tg  by HypSubst -3 -1
. . . | . . . . . | | |
. . . | . . . . . | | 1. x : Id
. . . | . . . . . | | 2. tg : Id
. . . | . . . . . | | 3. k : Knd
. . . | . . . . . | | 4. l : IdLnk
. . . | . . . . . | | 5. T : Type
. . . | . . . . . | | 6. A : Type
. . . | . . . . . | | 7. B : Type
. . . | . . . . . | | 8. f : A→B→(T List)
. . . | . . . . . | | 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | | 10. @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | |   ∈ Dsys
. . . | . . . . . | | 11. ∀D:Dsys.
. . . | . . . . . | | |   @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   D
. . . | . . . . . | | |   realizes es.(∀x@0:Id.
. . . | . . . . . | | |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . | | |               & (∀e:E.
. . . | . . . . . | | |                  loc(e) = source(l)
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |                   (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                    kind(e))))
. . . | . . . . . | | |               & (∀e:E.
. . . | . . . . . | | |                  isrcv(e)
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  lnk(e) = l
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |                   (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                    kind(e))))
. . . | . . . . . | | |               & (∀e:E.
. . . | . . . . . | | |                  loc(e) = source(l)
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  kind(e) = k
```

```
. . . | . . . . . | | |                  ⇒
. . . | . . . . . | | |                  ...)
. . . | . . . . . | | 12. @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | |   ∈ Dsys
. . . | . . . . . | | 13. D : Dsys
. . . | . . . . . | | 14. @source(l): ma-single-sends
. . . | . . . . . | | |   (x : A;
. . . | . . . . . | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |    k;
. . . | . . . . . | | |    l;
. . . | . . . . . | | |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | 15. ∀D':Dsys.
. . . | . . . . . | | |   D ⊆ D'
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (∀w:World, p:FairFifo.
. . . | . . . . . | | |    PossibleWorld(D';w)
. . . | . . . . . | | |    ⇒
. . . | . . . . . | | |    (∀x@0:Id.
. . . | . . . . . | | |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |    & (∀e:E.
. . . | . . . . . | | |       loc(e) = source(l)
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |    & (∀e:E.
. . . | . . . . . | | |       isrcv(e)
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       lnk(e) = l
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |    & (∀e:E.
. . . | . . . . . | | |       loc(e) = source(l)
. . . | . . . . . | | |       ⇒
. . . | . . . . . | | |       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);[<tg
```

```
. . . | . . . . . | | |                              ,λs,v. f
. . . | . . . . . | | |                                     (s(x)
. . . | . . . . . | | |                                     ,v)>])))))
. . . | . . . . . | | 16. D' : Dsys
. . . | . . . . . | | 17. D ⊆ D'
. . . | . . . . . | | 18. ∀w:World, p:FairFifo.
. . . | . . . . . | | |   PossibleWorld(D';w)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (∀x@0:Id.
. . . | . . . . . | | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      isrcv(e)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      lnk(e) = l
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      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);[<tg
. . . | . . . . . | | |                              ,λs,v. f
. . . | . . . . . | | |                                     (s(x)
. . . | . . . . . | | |                                     ,v)>])))
. . . | . . . . . | | 19. w : World
. . . | . . . . . | | 20. ∀p:FairFifo.
. . . | . . . . . | | |   PossibleWorld(D';w)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (∀x@0:Id.
. . . | . . . . . | | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
```

```
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      isrcv(e)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      lnk(e) = l
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |   & (∀e:E.
. . . | . . . . . | | |      loc(e) = source(l)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      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);[<tg
. . . | . . . . . | | |                             ,λs,v. f
. . . | . . . . . | | |                                    (s(x)
. . . | . . . . . | | |                                    ,v)>])))
. . . | . . . . . | | 21. p : FairFifo
. . . | . . . . . | | 22. PossibleWorld(D';w)
. . . | . . . . . | | 23. ∀x@0:Id.
. . . | . . . . . | | |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | | 24. ∀e:E.
. . . | . . . . . | | |   loc(e) = source(l)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | 25. ∀e:E.
. . . | . . . . . | | |   isrcv(e)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   lnk(e) = l
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | 26. ∀e:E.
. . . | . . . . . | | |   loc(e) = source(l)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   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);[<tg
. . . | . . . . . | | |                         ,λs,v. f(s(x),v)>]))
. . . | . . . . . | | 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | | 28. ∀e:E.
. . . | . . . . . | | |   loc(e) = source(l)
. . . | . . . . . | | |   ⇒
. . . | . . . . . | | |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | | 29. ∀e:E.
. . . | . . . . . | | |   kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | | 30. e : E
. . . | . . . . . | | 31. loc(e) = source(l)
. . . | . . . . . | | 32. kind(e) = k
. . . | . . . . . | | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | | 34. ∀e':E.
. . . | . . . . . | | |   (e' ∈ L)
. . . | . . . . . | | |   ⟺
. . . | . . . . . | | |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | | |   =
. . . | . . . . . | | |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | | 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | | 39. z :
. . . | . . . . . | | |   (tg@0:Id
. . . | . . . . . | | |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                rcv(l; tg@0))) List
. . . | . . . . . | | 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | 41. p1 :
. . . | . . . . . | | |   tg@0:Id
. . . | . . . . . | | |   ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |               rcv(l; tg@0))
. . . | . . . . . | | 42. (p1 ∈ map(λx@0.<tg,x@0>;f
. . . | . . . . . | | |                           ((x when e)
. . . | . . . . . | | |                           ,val(e))))
. . . | . . . . . | | ⊢  1of(p1) = tg
. . . | . . . . . | | | by RW0
. . . | . . . . . | | |    Thm* ∀a:T List, x:T', f:(T→T').
. . . | . . . . . | | |         (x ∈ map(f;a))
. . . | . . . . . | | |         ⟺
. . . | . . . . . | | |         (∃y:T. (y ∈ a) & x = f(y))
. . . | . . . . . | | |    {member_map}
. . . | . . . . . | | |    -1
. . . | . . . . . | | |
. . . | . . . . . | | |\
. . . | . . . . . | | | .....subterm .....
. . . | . . . . . | | | 1. x : Id
. . . | . . . . . | | | 2. tg : Id
. . . | . . . . . | | | 3. k : Knd
. . . | . . . . . | | | 4. l : IdLnk
```

```
. . . | . . . . . | | | 5. T : Type
. . . | . . . . . | | | 6. A : Type
. . . | . . . . . | | | 7. B : Type
. . . | . . . . . | | | 8. f : A→B→(T List)
. . . | . . . . . | | | 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | | | 10. @source(l): ma-single-sends
. . . | . . . . . | | | |   (x : A;
. . . | . . . . . | | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | | |    k;
. . . | . . . . . | | | |    l;
. . . | . . . . . | | | |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | | |   ∈ Dsys
. . . | . . . . . | | | 11. ∀D:Dsys.
. . . | . . . . . | | | |   @source(l): ma-single-sends
. . . | . . . . . | | | |   (x : A;
. . . | . . . . . | | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | | |    k;
. . . | . . . . . | | | |    l;
. . . | . . . . . | | | |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | | |   ⇒
. . . | . . . . . | | | |   D
. . . | . . . . . | | | |   realizes es....
. . . | . . . . . | | | |               & (∀e:E.
. . . | . . . . . | | | |                  loc(e) = source(l)
. . . | . . . . . | | | |                  ⇒
. . . | . . . . . | | | |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | | |                   (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | | |                    kind(e))))
. . . | . . . . . | | | |               & (∀e:E.
. . . | . . . . . | | | |                  isrcv(e)
. . . | . . . . . | | | |                  ⇒
. . . | . . . . . | | | |                  lnk(e) = l
. . . | . . . . . | | | |                  ⇒
. . . | . . . . . | | | |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | | |                   (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | | |                    kind(e))))
. . . | . . . . . | | | |               & (∀e:E.
. . . | . . . . . | | | |                  loc(e) = source(l)
. . . | . . . . . | | | |                  ⇒
. . . | . . . . . | | | |                  kind(e) = k
. . . | . . . . . | | | |                  ⇒
. . . | . . . . . | | | |                  ...)
. . . | . . . . . | | | 12. @source(l): ma-single-sends
. . . | . . . . . | | | |   (x : A;
. . . | . . . . . | | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | | |    k;
. . . | . . . . . | | | |    l;
. . . | . . . . . | | | |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | | |   ∈ Dsys
. . . | . . . . . | | | 13. D : Dsys
. . . | . . . . . | | | 14. @source(l): ma-single-sends
. . . | . . . . . | | | |   (x : A;
. . . | . . . . . | | | |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | | |    k;
. . . | . . . . . | | | |    l;
```

```
. . . | . . . . . | | | |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | | 15. ∀D':Dsys.
. . . | . . . . . | | | |   D ⊆ D'
. . . | . . . . . | | | |   ⇒
. . . | . . . . . | | | |   (∀w:World, p:FairFifo.
. . . | . . . . . | | | |    PossibleWorld(D';w)
. . . | . . . . . | | | |    ⇒
. . . | . . . . . | | | |    ...
. . . | . . . . . | | | |    & (∀e:E.
. . . | . . . . . | | | |       loc(e) = source(l)
. . . | . . . . . | | | |       ⇒
. . . | . . . . . | | | |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | | |    & (∀e:E.
. . . | . . . . . | | | |       isrcv(e)
. . . | . . . . . | | | |       ⇒
. . . | . . . . . | | | |       lnk(e) = l
. . . | . . . . . | | | |       ⇒
. . . | . . . . . | | | |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | | |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | | |    & (∀e:E.
. . . | . . . . . | | | |       loc(e) = source(l)
. . . | . . . . . | | | |       ⇒
. . . | . . . . . | | | |       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);[<tg
. . . | . . . . . | | | |                              ,λs,v. f
. . . | . . . . . | | | |                                     (s(x)
. . . | . . . . . | | | |                                     ,v)>])))))
. . . | . . . . . | | | 16. D' : Dsys
. . . | . . . . . | | | 17. D ⊆ D'
. . . | . . . . . | | | 18. ∀w:World, p:FairFifo.
. . . | . . . . . | | | |   PossibleWorld(D';w)
. . . | . . . . . | | | |   ⇒
. . . | . . . . . | | | |   (∀x@0:Id.
. . . | . . . . . | | | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | | |   & (∀e:E.
. . . | . . . . . | | | |      loc(e) = source(l)
. . . | . . . . . | | | |      ⇒
. . . | . . . . . | | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | | |   & (∀e:E.
```

```
. . . | . . . . . | | | |      isrcv(e)
. . . | . . . . . | | | |      ⇒
. . . | . . . . . | | | |      lnk(e) = l
. . . | . . . . . | | | |      ⇒
. . . | . . . . . | | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | | |   & (∀e:E.
. . . | . . . . . | | | |      loc(e) = source(l)
. . . | . . . . . | | | |      ⇒
. . . | . . . . . | | | |      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);[<tg
. . . | . . . . . | | | |                            ,λs,v. f
. . . | . . . . . | | | |                                   (s(x)
. . . | . . . . . | | | |                                   ,v)>])))
. . . | . . . . . | | | 19. w : World
. . . | . . . . . | | | 20. ∀p:FairFifo.
. . . | . . . . . | | | |   PossibleWorld(D';w)
. . . | . . . . . | | | |   ⇒

. . . | . . . . . | | | |   (∀x@0:Id.
. . . | . . . . . | | | |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | | |   & (∀e:E.
. . . | . . . . . | | | |      loc(e) = source(l)
. . . | . . . . . | | | |      ⇒
. . . | . . . . . | | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | | |   & (∀e:E.
. . . | . . . . . | | | |      isrcv(e)
. . . | . . . . . | | | |      ⇒
. . . | . . . . . | | | |      lnk(e) = l
. . . | . . . . . | | | |      ⇒
. . . | . . . . . | | | |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | | |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | | |   & (∀e:E.
. . . | . . . . . | | | |      loc(e) = source(l)
. . . | . . . . . | | | |      ⇒
. . . | . . . . . | | | |      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);[<tg
. . . | . . . . . | | | |                            ,λs,v. f
. . . | . . . . . | | | |                                   (s(x)
. . . | . . . . . | | | |                                   ,v)>])))
. . . | . . . . . | | | 21. p : FairFifo
. . . | . . . . . | | | 22. PossibleWorld(D';w)
. . . | . . . . . | | | 23. ∀x@0:Id.
. . . | . . . . . | | | |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | | | 24. ∀e:E.
. . . | . . . . . | | | |   loc(e) = source(l)
. . . | . . . . . | | | |   ⇒
. . . | . . . . . | | | |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | | |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | | 25. ∀e:E.
. . . | . . . . . | | | |   isrcv(e)
. . . | . . . . . | | | |   ⇒
. . . | . . . . . | | | |   lnk(e) = l
. . . | . . . . . | | | |   ⇒
. . . | . . . . . | | | |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | | |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | | 26. ∀e:E.
. . . | . . . . . | | | |   loc(e) = source(l)
. . . | . . . . . | | | |   ⇒
. . . | . . . . . | | | |   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);[<tg
. . . | . . . . . | | | |                          ,λs,v. f(s(x),v)>]))
. . . | . . . . . | | | 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | | | 28. ∀e:E.
. . . | . . . . . | | | |   loc(e) = source(l)
. . . | . . . . . | | | |   ⇒
. . . | . . . . . | | | |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | | | 29. ∀e:E.
. . . | . . . . . | | | |   kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
```

```
. . . | . . . . . | | | 30. e : E
. . . | . . . . . | | | 31. loc(e) = source(l)
. . . | . . . . . | | | 32. kind(e) = k
. . . | . . . . . | | | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | | | 34. ∀e':E.
. . . | . . . . . | | | |   (e' ∈ L)
. . . | . . . . . | | | |   ⟺
. . . | . . . . . | | | |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | | | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | | | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | | | |   =
. . . | . . . . . | | | |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | | | 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | | | 39. z :
. . . | . . . . . | | | |   (tg@0:Id
. . . | . . . . . | | | |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | | |                rcv(l; tg@0))) List
. . . | . . . . . | | | 40. z
. . . | . . . . . | | | |   =
. . . | . . . . . | | | |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | | 41. p1 :

. . . | . . . . . | | | |   tg@0:Id
. . . | . . . . . | | | |   ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | | |               rcv(l; tg@0))
. . . | . . . . . | | | 42. (p1 ∈ map(λx@0.<tg,x@0>;f
. . . | . . . . . | | | |                           ((x when e)
. . . | . . . . . | | | |                           ,val(e))))
. . . | . . . . . | | | 43. x@0 : T
. . . | . . . . . | | | ⊢  x@0
. . . | . . . . . | | | |  ∈ ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | | |               rcv(l; tg))
. . . | . . . . . | | | | by Unfold `ma-valtype` 0 THEN ReduceFpf 0
. . . | . . . . . | | | |    THEN
. . . | . . . . . | | | |    SplitOnConclITE
. . . | . . . . . | | | |
. . . | . . . . . | | | .....unproved Inclusion subgoal .....
. . . | . . . . . | | | 1. x : Id
. . . | . . . . . | | | 2. tg : Id
. . . | . . . . . | | | 3. k : Knd
. . . | . . . . . | | | 4. l : IdLnk
. . . | . . . . . | | | 5. T : Type
. . . | . . . . . | | | 6. A : Type
. . . | . . . . . | | | 7. B : Type
. . . | . . . . . | | | 8. f : A→B→(T List)
. . . | . . . . . | | | 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | | | 10. @source(l): ma-single-sends
. . . | . . . . . | | |     (x : A;
. . . | . . . . . | | |      k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |      k;
. . . | . . . . . | | |      l;
. . . | . . . . . | | |      [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | |     ∈ Dsys
. . . | . . . . . | | | 11. ∀D:Dsys.
. . . | . . . . . | | |     @source(l): ma-single-sends
```

```
. . . | . . . . . | | |     (x : A;
. . . | . . . . . | | |      k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |      k;
. . . | . . . . . | | |      l;
. . . | . . . . . | | |      [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | |     ⇒
. . . | . . . . . | | |     D
. . . | . . . . . | | |     realizes es....
. . . | . . . . . | | |                 & (∀e:E.
. . . | . . . . . | | |                    loc(e) = source(l)
. . . | . . . . . | | |                    ⇒
. . . | . . . . . | | |                    (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |                     (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                      kind(e))))
. . . | . . . . . | | |                 & (∀e:E.
. . . | . . . . . | | |                    isrcv(e)
. . . | . . . . . | | |                    ⇒
. . . | . . . . . | | |                    lnk(e) = l
. . . | . . . . . | | |                    ⇒
. . . | . . . . . | | |                    (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |                     (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                      kind(e))))
. . . | . . . . . | | |                 & (∀e:E.
. . . | . . . . . | | |                    loc(e) = source(l)
. . . | . . . . . | | |                    ⇒
. . . | . . . . . | | |                    kind(e) = k
. . . | . . . . . | | |                    ⇒
. . . | . . . . . | | |                    ...)
. . . | . . . . . | | | 12. @source(l): ma-single-sends
. . . | . . . . . | | |     (x : A;
. . . | . . . . . | | |      k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |      k;
. . . | . . . . . | | |      l;
. . . | . . . . . | | |      [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | |     ∈ Dsys
. . . | . . . . . | | | 13. D : Dsys
. . . | . . . . . | | | 14. @source(l): ma-single-sends
. . . | . . . . . | | |     (x : A;
. . . | . . . . . | | |      k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |      k;
. . . | . . . . . | | |      l;
. . . | . . . . . | | |      [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | | 15. ∀D':Dsys.
. . . | . . . . . | | |     D ⊆ D'
. . . | . . . . . | | |     ⇒
. . . | . . . . . | | |     (∀w:World, p:FairFifo.
. . . | . . . . . | | |      PossibleWorld(D';w)
. . . | . . . . . | | |      ⇒
. . . | . . . . . | | |      ...
. . . | . . . . . | | |      & (∀e:E.
. . . | . . . . . | | |         loc(e) = source(l)
. . . | . . . . . | | |         ⇒
. . . | . . . . . | | |         (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |      & (∀e:E.
```

```
. . . | . . . . . | | |          isrcv(e)
. . . | . . . . . | | |          ⇒
. . . | . . . . . | | |          lnk(e) = l
. . . | . . . . . | | |          ⇒
. . . | . . . . . | | |          (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |           (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |       & (∀e:E.
. . . | . . . . . | | |          loc(e) = source(l)
. . . | . . . . . | | |          ⇒
. . . | . . . . . | | |          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);[<tg
. . . | . . . . . | | |                                ,λs,v. f
. . . | . . . . . | | |                                       (s(x)
. . . | . . . . . | | |                                       ,v)>])))))
. . . | . . . . . | | | 16. D' : Dsys
. . . | . . . . . | | | 17. D ⊆ D'
. . . | . . . . . | | | 18. ∀w:World, p:FairFifo.
. . . | . . . . . | | |     PossibleWorld(D';w)
. . . | . . . . . | | |     ⇒
. . . | . . . . . | | |     (∀x@0:Id.
. . . | . . . . . | | |      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |     & (∀e:E.
. . . | . . . . . | | |        loc(e) = source(l)
. . . | . . . . . | | |        ⇒
. . . | . . . . . | | |        (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |     & (∀e:E.
. . . | . . . . . | | |        isrcv(e)
. . . | . . . . . | | |        ⇒
. . . | . . . . . | | |        lnk(e) = l
. . . | . . . . . | | |        ⇒
. . . | . . . . . | | |        (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |     & (∀e:E.
. . . | . . . . . | | |        loc(e) = source(l)
. . . | . . . . . | | |        ⇒
. . . | . . . . . | | |        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);[<tg
. . . | . . . . . | | |                                ,λs,v. f
. . . | . . . . . | | |                                       (s(x)
. . . | . . . . . | | |                                       ,v)>])))
. . . | . . . . . | | | 19. w : World
. . . | . . . . . | | | 20. ∀p:FairFifo.
. . . | . . . . . | | |     PossibleWorld(D';w)
. . . | . . . . . | | |     ⇒
. . . | . . . . . | | |     (∀x@0:Id.
. . . | . . . . . | | |      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | |     & (∀e:E.
. . . | . . . . . | | |        loc(e) = source(l)
. . . | . . . . . | | |        ⇒
. . . | . . . . . | | |        (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |     & (∀e:E.
. . . | . . . . . | | |        isrcv(e)
. . . | . . . . . | | |        ⇒
. . . | . . . . . | | |        lnk(e) = l
. . . | . . . . . | | |        ⇒
. . . | . . . . . | | |        (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | |     & (∀e:E.
. . . | . . . . . | | |        loc(e) = source(l)
. . . | . . . . . | | |        ⇒
. . . | . . . . . | | |        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);[<tg
. . . | . . . . . | | |                                ,λs,v. f
. . . | . . . . . | | |                                       (s(x)
. . . | . . . . . | | |                                       ,v)>])))
. . . | . . . . . | | | 21. p : FairFifo
```

```
. . . | . . . . . | | | 22. PossibleWorld(D';w)
. . . | . . . . . | | | 23. ∀x@0:Id.
. . . | . . . . . | | |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | | | 24. ∀e:E.
. . . | . . . . . | | |     loc(e) = source(l)
. . . | . . . . . | | |     ⇒
. . . | . . . . . | | |     (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | | 25. ∀e:E.
. . . | . . . . . | | |     isrcv(e)
. . . | . . . . . | | |     ⇒
. . . | . . . . . | | |     lnk(e) = l
. . . | . . . . . | | |     ⇒
. . . | . . . . . | | |     (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | |      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | | 26. ∀e:E.
. . . | . . . . . | | |     loc(e) = source(l)
. . . | . . . . . | | |     ⇒
. . . | . . . . . | | |     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);[<tg
. . . | . . . . . | | |                           ,λs,v. f(s(x),v)>]))
. . . | . . . . . | | | 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | | | 28. ∀e:E.
. . . | . . . . . | | |     loc(e) = source(l)
. . . | . . . . . | | |     ⇒
. . . | . . . . . | | |     kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | | | 29. ∀e:E.
. . . | . . . . . | | |     kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | | | 30. e : E
. . . | . . . . . | | | 31. loc(e) = source(l)
. . . | . . . . . | | | 32. kind(e) = k
. . . | . . . . . | | | 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | | | 34. ∀e':E.
. . . | . . . . . | | |     (e' ∈ L)
. . . | . . . . . | | |     ⟺
. . . | . . . . . | | |     isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | | | 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | | | 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | | |     =
. . . | . . . . . | | |     map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | | 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | | | 38. f((x when e),val(e)) ∈ T List
```

```
. . . | . . . . . | | | 39. z :
. . . | . . . . . | | |     (tg@0:Id
. . . | . . . . . | | |      ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                  rcv(l; tg@0))) List
. . . | . . . . . | | | 40. z
. . . | . . . . . | | |     =
. . . | . . . . . | | |     map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | | 41. p1 :
. . . | . . . . . | | |     tg@0:Id
. . . | . . . . . | | |     ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | |                 rcv(l; tg@0))
. . . | . . . . . | | | 42. (p1 ∈ map(λx@0.<tg,x@0>;f
. . . | . . . . . | | |                              ((x when e)
. . . | . . . . . | | |                              ,val(e))))
. . . | . . . . . | | | 43. x@0 : T
. . . | . . . . . | | | 44. k = rcv(l; tg)
. . . | . . . . . | | | ⊢  x@0 ∈ B
. . . | . . . . . | | |   by Assert (T = B) THEN BackThruSomeHyp
. . . | . . . . . | | |      THEN
. . . | . . . . . | | |      RWO
. . . | . . . . . | | |      Thm* ∀a,b:Knd. a = b ⟺ a = b
. . . | . . . . . | | |      {assert_DASH_eq_DASH_knd}
. . . | . . . . . | | |      -1
. . . | . . . . . | | |
. . . | . . . . . | | .\
. . . | . . . . . | | . 1. x : Id
. . . | . . . . . | | . 2. tg : Id
. . . | . . . . . | | . 3. k : Knd
. . . | . . . . . | | . 4. l : IdLnk
. . . | . . . . . | | . 5. T : Type
. . . | . . . . . | | . 6. A : Type
. . . | . . . . . | | . 7. B : Type
. . . | . . . . . | | . 8. f : A→B→(T List)
. . . | . . . . . | | . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | | . 10. @source(l): ma-single-sends
. . . | . . . . . | | .     (x : A;
. . . | . . . . . | | .      k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | .      k;
. . . | . . . . . | | .      l;
. . . | . . . . . | | .      [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | .     ∈ Dsys
. . . | . . . . . | | . 11. ∀D:Dsys.
. . . | . . . . . | | .     @source(l): ma-single-sends
. . . | . . . . . | | .     (x : A;
. . . | . . . . . | | .      k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | .      k;
. . . | . . . . . | | .      l;
. . . | . . . . . | | .      [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | .     ⇒
. . . | . . . . . | | .     D
. . . | . . . . . | | .     realizes es....
. . . | . . . . . | | .                 & (∀e:E.
. . . | . . . . . | | .                    loc(e) = source(l)
. . . | . . . . . | | .                    ⇒
. . . | . . . . . | | .                    (valtype(e) ⊆r ma-valtype
```

```
. . . | . . . . . | | .                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | .                     kind(e))))
. . . | . . . . . | | .                & (∀e:E.
. . . | . . . . . | | .                   isrcv(e)
. . . | . . . . . | | .                   ⇒
. . . | . . . . . | | .                   lnk(e) = l
. . . | . . . . . | | .                   ⇒
. . . | . . . . . | | .                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | .                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | .                     kind(e))))
. . . | . . . . . | | .                & (∀e:E.
. . . | . . . . . | | .                   loc(e) = source(l)
. . . | . . . . . | | .                   ⇒
. . . | . . . . . | | .                   kind(e) = k
. . . | . . . . . | | .                   ⇒
. . . | . . . . . | | .                   ...)
. . . | . . . . . | | . 12. @source(l): ma-single-sends
. . . | . . . . . | | .     (x : A;
. . . | . . . . . | | .      k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | .      k;
. . . | . . . . . | | .      l;
. . . | . . . . . | | .      [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | | .     ∈ Dsys
. . . | . . . . . | | . 13. D : Dsys
. . . | . . . . . | | . 14. @source(l): ma-single-sends
. . . | . . . . . | | .     (x : A;
. . . | . . . . . | | .      k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | .      k;
. . . | . . . . . | | .      l;
. . . | . . . . . | | .      [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | | . 15. ∀D':Dsys.
. . . | . . . . . | | .     D ⊆ D'
. . . | . . . . . | | .     ⇒
. . . | . . . . . | | .     (∀w:World, p:FairFifo.
. . . | . . . . . | | .      PossibleWorld(D';w)
. . . | . . . . . | | .      ⇒
. . . | . . . . . | | .      ...
. . . | . . . . . | | .      & (∀e:E.
. . . | . . . . . | | .         loc(e) = source(l)
. . . | . . . . . | | .         ⇒
. . . | . . . . . | | .         (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | .          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | .      & (∀e:E.
. . . | . . . . . | | .         isrcv(e)
. . . | . . . . . | | .         ⇒
. . . | . . . . . | | .         lnk(e) = l
. . . | . . . . . | | .         ⇒
. . . | . . . . . | | .         (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | .          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | .      & (∀e:E.
. . . | . . . . . | | .         loc(e) = source(l)
. . . | . . . . . | | .         ⇒
. . . | . . . . . | | .         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);[<tg
. . . | . . . . . | | .                              ,λs,v. f
. . . | . . . . . | | .                                     (s(x)
. . . | . . . . . | | .                                     ,v)>])))))
. . . | . . . . . | | . 16. D' : Dsys
. . . | . . . . . | | . 17. D ⊆ D'
. . . | . . . . . | | . 18. ∀w:World, p:FairFifo.
. . . | . . . . . | | .     PossibleWorld(D';w)
. . . | . . . . . | | .     ⇒
. . . | . . . . . | | .     (∀x@0:Id.
. . . | . . . . . | | .      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | .     & (∀e:E.
. . . | . . . . . | | .        loc(e) = source(l)
. . . | . . . . . | | .        ⇒
. . . | . . . . . | | .        (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | .     & (∀e:E.
. . . | . . . . . | | .        isrcv(e)
. . . | . . . . . | | .        ⇒
. . . | . . . . . | | .        lnk(e) = l
. . . | . . . . . | | .        ⇒
. . . | . . . . . | | .        (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | .     & (∀e:E.
. . . | . . . . . | | .        loc(e) = source(l)
. . . | . . . . . | | .        ⇒
. . . | . . . . . | | .        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);[<tg
. . . | . . . . . | | .                              ,λs,v. f
```

```
. . . | . . . . . | | .                                    (s(x)
. . . | . . . . . | | .                                    ,v)>])))
. . . | . . . . . | | . 19. w : World
. . . | . . . . . | | . 20. ∀p:FairFifo.
. . . | . . . . . | | .     PossibleWorld(D';w)
. . . | . . . . . | | .     ⇒
. . . | . . . . . | | .     (∀x@0:Id.
. . . | . . . . . | | .      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | | .     & (∀e:E.
. . . | . . . . . | | .        loc(e) = source(l)
. . . | . . . . . | | .        ⇒
. . . | . . . . . | | .        (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | .     & (∀e:E.
. . . | . . . . . | | .        isrcv(e)
. . . | . . . . . | | .        ⇒
. . . | . . . . . | | .        lnk(e) = l
. . . | . . . . . | | .        ⇒
. . . | . . . . . | | .        (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | | .     & (∀e:E.
. . . | . . . . . | | .        loc(e) = source(l)
. . . | . . . . . | | .        ⇒
. . . | . . . . . | | .        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);[<tg
. . . | . . . . . | | .                              ,λs,v. f
. . . | . . . . . | | .                                     (s(x)
. . . | . . . . . | | .                                     ,v)>])))
. . . | . . . . . | | . 21. p : FairFifo
. . . | . . . . . | | . 22. PossibleWorld(D';w)
. . . | . . . . . | | . 23. ∀x@0:Id.
. . . | . . . . . | | .     vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | | . 24. ∀e:E.
. . . | . . . . . | | .     loc(e) = source(l)
. . . | . . . . . | | .     ⇒
. . . | . . . . . | | .     (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | .      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | . 25. ∀e:E.
. . . | . . . . . | | .     isrcv(e)
. . . | . . . . . | | .     ⇒
. . . | . . . . . | | .     lnk(e) = l
```

```
. . . | . . . . . | | .     ⇒
. . . | . . . . . | | .     (valtype(e) ⊆r ma-valtype
. . . | . . . . . | | .      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | | . 26. ∀e:E.
. . . | . . . . . | | .     loc(e) = source(l)
. . . | . . . . . | | .     ⇒
. . . | . . . . . | | .     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);[<tg
. . . | . . . . . | | .                           ,λs,v. f(s(x),v)>]))
. . . | . . . . . | | . 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | | . 28. ∀e:E.
. . . | . . . . . | | .     loc(e) = source(l)
. . . | . . . . . | | .     ⇒
. . . | . . . . . | | .     kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | | . 29. ∀e:E.
. . . | . . . . . | | .     kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | | . 30. e : E
. . . | . . . . . | | . 31. loc(e) = source(l)
. . . | . . . . . | | . 32. kind(e) = k
. . . | . . . . . | | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | | . 34. ∀e':E.
. . . | . . . . . | | .     (e' ∈ L)
. . . | . . . . . | | .     ⟺
. . . | . . . . . | | .     isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | | . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | | .     =
. . . | . . . . . | | .     map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | | . 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | | . 39. z :
. . . | . . . . . | | .     (tg@0:Id
. . . | . . . . . | | .      ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | .                  rcv(l; tg@0))) List
. . . | . . . . . | | . 40. z
. . . | . . . . . | | .     =
. . . | . . . . . | | .     map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | | . 41. p1 :
. . . | . . . . . | | .     tg@0:Id
. . . | . . . . . | | .     ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | | .                 rcv(l; tg@0))
. . . | . . . . . | | . 42. ∃y:T.
```

```
. . . | . . . . . | | .     (y ∈ f((x when e),val(e)))
. . . | . . . . . | | .     & p1 = (λx@0.<tg,x@0>)(y)
. . . | . . . . . | | . ⊢  1of(p1) = tg
. . . | . . . . . | | .   by (Reduce -1 THEN ExRepD
. . . | . . . . . | | .       THEN
. . . | . . . . . | | .       HypSubst -1 0)
. . . | . . . . . | | .      THEN
. . . | . . . . . | | .      (Reduce 0)
. . . | . . . . . | |
. . . | . . . . . | .\
. . . | . . . . . | . 1. x : Id
. . . | . . . . . | . 2. tg : Id
. . . | . . . . . | . 3. k : Knd
. . . | . . . . . | . 4. l : IdLnk
. . . | . . . . . | . 5. T : Type
. . . | . . . . . | . 6. A : Type
. . . | . . . . . | . 7. B : Type
. . . | . . . . . | . 8. f : A→B→(T List)
. . . | . . . . . | . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | . 10. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | . |   ∈ Dsys
. . . | . . . . . | . 11. ∀D:Dsys.
. . . | . . . . . | . |   @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   D
. . . | . . . . . | . |   realizes es.(∀x@0:Id.
. . . | . . . . . | . |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                   loc(e) = source(l)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     kind(e))))
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                   isrcv(e)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   lnk(e) = l
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     kind(e))))
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                   loc(e) = source(l)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   kind(e) = k
```

```
. . . | . . . . . | . |                  ⇒
. . . | . . . . . | . |                  ...)
. . . | . . . . . | . 12. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | . |   ∈ Dsys
. . . | . . . . . | . 13. D : Dsys
. . . | . . . . . | . 14. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . 15. ∀D':Dsys.
. . . | . . . . . | . |   D ⊆ D'
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀w:World, p:FairFifo.
. . . | . . . . . | . |    PossibleWorld(D';w)
. . . | . . . . . | . |    ⇒
. . . | . . . . . | . |    (∀x@0:Id.
. . . | . . . . . | . |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       loc(e) = source(l)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       isrcv(e)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       lnk(e) = l
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       loc(e) = source(l)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       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);[<tg
```

```
. . . | . . . . . | . |                              ,λs,v. f
. . . | . . . . . | . |                                     (s(x)
. . . | . . . . . | . |                                     ,v)>])))))
. . . | . . . . . | . 16. D' : Dsys
. . . | . . . . . | . 17. D ⊆ D'
. . . | . . . . . | . 18. ∀w:World, p:FairFifo.
. . . | . . . . . | . |   PossibleWorld(D';w)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀x@0:Id.
. . . | . . . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      isrcv(e)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      lnk(e) = l
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      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);[<tg
. . . | . . . . . | . |                              ,λs,v. f
. . . | . . . . . | . |                                     (s(x)
. . . | . . . . . | . |                                     ,v)>])))
. . . | . . . . . | . 19. w : World
. . . | . . . . . | . 20. ∀p:FairFifo.
. . . | . . . . . | . |   PossibleWorld(D';w)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀x@0:Id.
. . . | . . . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
```

```
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      isrcv(e)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      lnk(e) = l
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      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);[<tg
. . . | . . . . . | . |                              ,λs,v. f
. . . | . . . . . | . |                                     (s(x)
. . . | . . . . . | . |                                     ,v)>])))
. . . | . . . . . | . 21. p : FairFifo
. . . | . . . . . | . 22. PossibleWorld(D';w)
. . . | . . . . . | . 23. ∀x@0:Id.
. . . | . . . . . | . |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | . 24. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 25. ∀e:E.
. . . | . . . . . | . |   isrcv(e)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   lnk(e) = l
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 26. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   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);[<tg
. . . | . . . . . | . |                          ,λs,v. f(s(x),v)>]))
. . . | . . . . . | . 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | . 28. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | . 29. ∀e:E.
. . . | . . . . . | . |   kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | . 30. e : E
. . . | . . . . . | . 31. loc(e) = source(l)
. . . | . . . . . | . 32. kind(e) = k
. . . | . . . . . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | . 34. ∀e':E.
. . . | . . . . . | . |   (e' ∈ L)
. . . | . . . . . | . |   ⟺
. . . | . . . . . | . |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | . |   =
. . . | . . . . . | . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | . 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | . 39. z :
. . . | . . . . . | . |   (tg@0:Id
. . . | . . . . . | . |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                rcv(l; tg@0))) List
. . . | . . . . . | . 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 41. z
. . . | . . . . . | . |   ∈ {p:(tg@0:Id
. . . | . . . . . | . |         ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                      rcv(l; tg@0)))
. . . | . . . . . | . |      | 1of(p) = tg } List
. . . | . . . . . | . ⊢  z ∈ (Id×T) List  by DoSubsume
. . . | . . . . . | . |
. . . | . . . . . | . .....subtype_rel .....
. . . | . . . . . | . 1. x : Id
. . . | . . . . . | . 2. tg : Id
. . . | . . . . . | . 3. k : Knd
. . . | . . . . . | . 4. l : IdLnk
. . . | . . . . . | . 5. T : Type
. . . | . . . . . | . 6. A : Type
. . . | . . . . . | . 7. B : Type
. . . | . . . . . | . 8. f : A→B→(T List)
. . . | . . . . . | . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | . 10. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
```

```
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | . |   ∈ Dsys
. . . | . . . . . | . 11. ∀D:Dsys.
. . . | . . . . . | . |   @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   D
. . . | . . . . . | . |   realizes es.(∀x@0:Id.
. . . | . . . . . | . |                 vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . | . |                & (∀e:E.
. . . | . . . . . | . |                   loc(e) = source(l)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     kind(e))))
. . . | . . . . . | . |                & (∀e:E.
. . . | . . . . . | . |                   isrcv(e)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   lnk(e) = l
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     kind(e))))
. . . | . . . . . | . |                & (∀e:E.
. . . | . . . . . | . |                   loc(e) = source(l)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   kind(e) = k
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   ...)
. . . | . . . . . | . 12. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | . |   ∈ Dsys
. . . | . . . . . | . 13. D : Dsys
. . . | . . . . . | . 14. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . 15. ∀D':Dsys.
. . . | . . . . . | . |   D ⊆ D'
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀w:World, p:FairFifo.
. . . | . . . . . | . |    PossibleWorld(D';w)
. . . | . . . . . | . |    ⇒
. . . | . . . . . | . |    (∀x@0:Id.
. . . | . . . . . | . |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
```

```
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       loc(e) = source(l)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       isrcv(e)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       lnk(e) = l
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       loc(e) = source(l)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       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);[<tg
. . . | . . . . . | . |                                ,λs,v. f
. . . | . . . . . | . |                                       (s(x)
. . . | . . . . . | . |                                       ,v)>])))))
. . . | . . . . . | . 16. D' : Dsys
. . . | . . . . . | . 17. D ⊆ D'
. . . | . . . . . | . 18. ∀w:World, p:FairFifo.
. . . | . . . . . | . |   PossibleWorld(D';w)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀x@0:Id.
. . . | . . . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      isrcv(e)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      lnk(e) = l
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
```

```
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      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);[<tg
. . . | . . . . . | . |                            ,λs,v. f
. . . | . . . . . | . |                                   (s(x)
. . . | . . . . . | . |                                   ,v)>])))
. . . | . . . . . | . 19. w : World
. . . | . . . . . | . 20. ∀p:FairFifo.
. . . | . . . . . | . |   PossibleWorld(D';w)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀x@0:Id.
. . . | . . . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      isrcv(e)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      lnk(e) = l
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      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);[<tg
. . . | . . . . . | . |                           ,λs,v. f
. . . | . . . . . | . |                                  (s(x)
. . . | . . . . . | . |                                  ,v)>])))
. . . | . . . . . | . 21. p : FairFifo
. . . | . . . . . | . 22. PossibleWorld(D';w)
. . . | . . . . . | . 23. ∀x@0:Id.
. . . | . . . . . | . |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | . 24. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 25. ∀e:E.
. . . | . . . . . | . |   isrcv(e)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   lnk(e) = l
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 26. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   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);[<tg
. . . | . . . . . | . |                           ,λs,v. f(s(x),v)>]))
. . . | . . . . . | . 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | . 28. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | . 29. ∀e:E.
. . . | . . . . . | . |   kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | . 30. e : E
. . . | . . . . . | . 31. loc(e) = source(l)
. . . | . . . . . | . 32. kind(e) = k
. . . | . . . . . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | . 34. ∀e':E.
. . . | . . . . . | . |   (e' ∈ L)
. . . | . . . . . | . |   ⟺
. . . | . . . . . | . |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | . 36. map(λe'.<tag(e'),val(e')>;L)
```

```
. . . | . . . . . | . |   =
. . . | . . . . . | . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | . 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | . 39. z :
. . . | . . . . . | . |   (tg@0:Id
. . . | . . . . . | . |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                rcv(l; tg@0))) List
. . . | . . . . . | . 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 41. z
. . . | . . . . . | . |   ∈ {p:(tg@0:Id
. . . | . . . . . | . |         ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                      rcv(l; tg@0)))
. . . | . . . . . | . |      | 1of(p) = tg } List
. . . | . . . . . | . ⊢  ({p:(tg@0:Id
. . . | . . . . . | . |       ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                    rcv(l; tg@0)))
. . . | . . . . . | . |    | 1of(p) = tg } List) ⊆r ((Id×T) List)
. . . | . . . . . | . | by UseSubtypeRelLemmas
. . . | . . . . . | . |
. . . | . . . . . | . 1. x : Id
. . . | . . . . . | . 2. tg : Id
. . . | . . . . . | . 3. k : Knd
. . . | . . . . . | . 4. l : IdLnk
. . . | . . . . . | . 5. T : Type
. . . | . . . . . | . 6. A : Type
. . . | . . . . . | . 7. B : Type
. . . | . . . . . | . 8. f : A→B→(T List)
. . . | . . . . . | . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | . 10. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | . |   ∈ Dsys
. . . | . . . . . | . 11. ∀D:Dsys.
. . . | . . . . . | . |   @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   D
. . . | . . . . . | . |   realizes es.(∀x@0:Id.
. . . | . . . . . | . |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                   loc(e) = source(l)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     kind(e))))
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                   isrcv(e)
```

```
. . . | . . . . . | . |                  ⇒
. . . | . . . . . | . |                  lnk(e) = l
. . . | . . . . . | . |                  ⇒
. . . | . . . . . | . |                  (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |                   (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                    kind(e))))
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                  loc(e) = source(l)
. . . | . . . . . | . |                  ⇒
. . . | . . . . . | . |                  kind(e) = k
. . . | . . . . . | . |                  ⇒
. . . | . . . . . | . |                  ...)
. . . | . . . . . | . 12. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | . |   ∈ Dsys
. . . | . . . . . | . 13. D : Dsys
. . . | . . . . . | . 14. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;

. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . 15. ∀D':Dsys.
. . . | . . . . . | . |   D ⊆ D'
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀w:World, p:FairFifo.
. . . | . . . . . | . |    PossibleWorld(D';w)
. . . | . . . . . | . |    ⇒
. . . | . . . . . | . |    (∀x@0:Id.
. . . | . . . . . | . |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       loc(e) = source(l)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       isrcv(e)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       lnk(e) = l
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       loc(e) = source(l)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       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);[<tg
. . . | . . . . . | . |                              ,λs,v. f
. . . | . . . . . | . |                                     (s(x)
. . . | . . . . . | . |                                     ,v)>])))))
. . . | . . . . . | . 16. D' : Dsys
. . . | . . . . . | . 17. D ⊆ D'
. . . | . . . . . | . 18. ∀w:World, p:FairFifo.
. . . | . . . . . | . |   PossibleWorld(D';w)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀x@0:Id.
. . . | . . . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      isrcv(e)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      lnk(e) = l
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      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);[<tg
. . . | . . . . . | . |                              ,λs,v. f
. . . | . . . . . | . |                                     (s(x)
. . . | . . . . . | . |                                     ,v)>])))
. . . | . . . . . | . 19. w : World
```

```
. . . | . . . . . | . 20. ∀p:FairFifo.
. . . | . . . . . | . |   PossibleWorld(D';w)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀x@0:Id.
. . . | . . . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      isrcv(e)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      lnk(e) = l
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      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);[<tg
. . . | . . . . . | . |                             ,λs,v. f
. . . | . . . . . | . |                                    (s(x)
. . . | . . . . . | . |                                    ,v)>])))
. . . | . . . . . | . 21. p : FairFifo
. . . | . . . . . | . 22. PossibleWorld(D';w)
. . . | . . . . . | . 23. ∀x@0:Id.
. . . | . . . . . | . |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | . 24. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 25. ∀e:E.
. . . | . . . . . | . |   isrcv(e)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   lnk(e) = l
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
```

```
. . . | . . . . . | . 26. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   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);[<tg
. . . | . . . . . | . |                         ,λs,v. f(s(x),v)>]))
. . . | . . . . . | . 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | . 28. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | . 29. ∀e:E.
. . . | . . . . . | . |   kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | . 30. e : E
. . . | . . . . . | . 31. loc(e) = source(l)
. . . | . . . . . | . 32. kind(e) = k
. . . | . . . . . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | . 34. ∀e':E.
. . . | . . . . . | . |   (e' ∈ L)
. . . | . . . . . | . |   ⟺
. . . | . . . . . | . |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | . |   =
. . . | . . . . . | . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | . 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | . 39. z :
. . . | . . . . . | . |   (tg@0:Id
. . . | . . . . . | . |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                rcv(l; tg@0))) List
. . . | . . . . . | . 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 41. z
. . . | . . . . . | . |   ∈ {p:(tg@0:Id
. . . | . . . . . | . |         ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     rcv(l; tg@0)))
. . . | . . . . . | . |      | 1of(p) = tg } List
. . . | . . . . . | . ⊢  {p:(tg@0:Id
. . . | . . . . . | . |      ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                  rcv(l; tg@0)))
. . . | . . . . . | . |   | 1of(p) = tg } ⊆r (Id×T)
. . . | . . . . . | . | by Analyze 0 THEN Analyze -1 THEN Analyze -2
. . . | . . . . . | . |    THEN
. . . | . . . . . | . |    All Reduce
```

```
. . . | . . . . . | . |    THEN
. . . | . . . . . | . |    DoSubsume

. . . | . . . . . | . |
. . . | . . . . . | . .....subtype_rel .....
. . . | . . . . . | . 1. x : Id
. . . | . . . . . | . 2. tg : Id
. . . | . . . . . | . 3. k : Knd
. . . | . . . . . | . 4. l : IdLnk
. . . | . . . . . | . 5. T : Type
. . . | . . . . . | . 6. A : Type
. . . | . . . . . | . 7. B : Type
. . . | . . . . . | . 8. f : A→B→(T List)
. . . | . . . . . | . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | . 10. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | . |   ∈ Dsys
. . . | . . . . . | . 11. ∀D:Dsys.
. . . | . . . . . | . |   @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   D
. . . | . . . . . | . |   realizes es.(∀x@0:Id.
. . . | . . . . . | . |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                   loc(e) = source(l)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     kind(e))))
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                   isrcv(e)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   lnk(e) = l
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     kind(e))))
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                   loc(e) = source(l)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   kind(e) = k
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   ...)
. . . | . . . . . | . 12. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
```

```
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | . |   ∈ Dsys
. . . | . . . . . | . 13. D : Dsys
. . . | . . . . . | . 14. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . 15. ∀D':Dsys.
. . . | . . . . . | . |   D ⊆ D'
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀w:World, p:FairFifo.
. . . | . . . . . | . |    PossibleWorld(D';w)
. . . | . . . . . | . |    ⇒
. . . | . . . . . | . |    (∀x@0:Id.
. . . | . . . . . | . |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       loc(e) = source(l)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       isrcv(e)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       lnk(e) = l
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       loc(e) = source(l)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       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);[<tg
. . . | . . . . . | . |                              ,λs,v. f
. . . | . . . . . | . |                                     (s(x)
. . . | . . . . . | . |                                     ,v)>])))))
. . . | . . . . . | . 16. D' : Dsys
. . . | . . . . . | . 17. D ⊆ D'
. . . | . . . . . | . 18. ∀w:World, p:FairFifo.
```

```
. . . | . . . . . | . |   PossibleWorld(D';w)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀x@0:Id.
. . . | . . . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      isrcv(e)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      lnk(e) = l
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      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);[<tg
. . . | . . . . . | . |                            ,λs,v. f
. . . | . . . . . | . |                                   (s(x)
. . . | . . . . . | . |                                   ,v)>])))
. . . | . . . . . | . 19. w : World
. . . | . . . . . | . 20. ∀p:FairFifo.
. . . | . . . . . | . |   PossibleWorld(D';w)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀x@0:Id.
. . . | . . . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      isrcv(e)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      lnk(e) = l
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
```

```
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      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);[<tg
. . . | . . . . . | . |                            ,λs,v. f
. . . | . . . . . | . |                                   (s(x)
. . . | . . . . . | . |                                   ,v)>])))
. . . | . . . . . | . 21. p : FairFifo
. . . | . . . . . | . 22. PossibleWorld(D';w)
. . . | . . . . . | . 23. ∀x@0:Id.
. . . | . . . . . | . |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | . 24. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 25. ∀e:E.
. . . | . . . . . | . |   isrcv(e)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   lnk(e) = l
. . . | . . . . . | . |   ⇒

. . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 26. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   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);[<tg
```

```
. . . | . . . . . | . |                          ,λs,v. f(s(x),v)>]))
. . . | . . . . . | . 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | . 28. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | . 29. ∀e:E.
. . . | . . . . . | . |   kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | . 30. e : E
. . . | . . . . . | . 31. loc(e) = source(l)
. . . | . . . . . | . 32. kind(e) = k
. . . | . . . . . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | . 34. ∀e':E.
. . . | . . . . . | . |   (e' ∈ L)
. . . | . . . . . | . |   ⟺
. . . | . . . . . | . |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | . |   =
. . . | . . . . . | . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | . 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | . 39. z :
. . . | . . . . . | . |   (tg@0:Id
. . . | . . . . . | . |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                rcv(l; tg@0))) List
. . . | . . . . . | . 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 41. z
. . . | . . . . . | . |   ∈ {p:(tg@0:Id
. . . | . . . . . | . |         ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     rcv(l; tg@0)))
. . . | . . . . . | . |      | 1of(p) = tg } List
. . . | . . . . . | . 42. tg@0 : Id
. . . | . . . . . | . 43. ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |              rcv(l; tg@0))
. . . | . . . . . | . 44. tg@0 = tg
. . . | . . . . . | . ⊢  ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |             rcv(l; tg@0)) ⊆r T
. . . | . . . . . | . | by HypSubst' -1 0 THEN Unfold `ma-valtype` 0
. . . | . . . . . | . |    THEN
. . . | . . . . . | . |    ReduceFpf 0
. . . | . . . . . | . |
. . . | . . . . . | . 1. x : Id
. . . | . . . . . | . 2. tg : Id
. . . | . . . . . | . 3. k : Knd
. . . | . . . . . | . 4. l : IdLnk
. . . | . . . . . | . 5. T : Type
. . . | . . . . . | . 6. A : Type
. . . | . . . . . | . 7. B : Type
. . . | . . . . . | . 8. f : A→B→(T List)
. . . | . . . . . | . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | . 10. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
```

```
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | . |   ∈ Dsys
. . . | . . . . . | . 11. ∀D:Dsys.
. . . | . . . . . | . |   @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   D
. . . | . . . . . | . |   realizes es.(∀x@0:Id.
. . . | . . . . . | . |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                   loc(e) = source(l)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     kind(e))))
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                   isrcv(e)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   lnk(e) = l
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     kind(e))))
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                   loc(e) = source(l)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   kind(e) = k
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   ...)
. . . | . . . . . | . 12. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | . |   ∈ Dsys
. . . | . . . . . | . 13. D : Dsys
. . . | . . . . . | . 14. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . 15. ∀D':Dsys.
. . . | . . . . . | . |   D ⊆ D'
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀w:World, p:FairFifo.
. . . | . . . . . | . |    PossibleWorld(D';w)
. . . | . . . . . | . |    ⇒
. . . | . . . . . | . |    (∀x@0:Id.
```

```
. . . | . . . . . | . |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       loc(e) = source(l)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       isrcv(e)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       lnk(e) = l
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       loc(e) = source(l)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       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);[<tg
. . . | . . . . . | . |                              ,λs,v. f
. . . | . . . . . | . |                                     (s(x)
. . . | . . . . . | . |                                     ,v)>])))))
. . . | . . . . . | . 16. D' : Dsys
. . . | . . . . . | . 17. D ⊆ D'
. . . | . . . . . | . 18. ∀w:World, p:FairFifo.
. . . | . . . . . | . |   PossibleWorld(D';w)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀x@0:Id.
. . . | . . . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      isrcv(e)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      lnk(e) = l
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
```

```
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      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);[<tg
. . . | . . . . . | . |                           ,λs,v. f
. . . | . . . . . | . |                                  (s(x)
. . . | . . . . . | . |                                  ,v)>])))
. . . | . . . . . | . 19. w : World
. . . | . . . . . | . 20. ∀p:FairFifo.
. . . | . . . . . | . |   PossibleWorld(D';w)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀x@0:Id.
. . . | . . . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      isrcv(e)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      lnk(e) = l
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      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);[<tg
. . . | . . . . . | . |                           ,λs,v. f
. . . | . . . . . | . |                                  (s(x)
. . . | . . . . . | . |                                  ,v)>])))
. . . | . . . . . | . 21. p : FairFifo
. . . | . . . . . | . 22. PossibleWorld(D';w)
. . . | . . . . . | . 23. ∀x@0:Id.
. . . | . . . . . | . |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | . 24. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 25. ∀e:E.
. . . | . . . . . | . |   isrcv(e)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   lnk(e) = l
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 26. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   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);[<tg
. . . | . . . . . | . |                         ,λs,v. f(s(x),v)>]))
. . . | . . . . . | . 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | . 28. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | . 29. ∀e:E.
. . . | . . . . . | . |   kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | . 30. e : E
. . . | . . . . . | . 31. loc(e) = source(l)
. . . | . . . . . | . 32. kind(e) = k
. . . | . . . . . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | . 34. ∀e':E.
. . . | . . . . . | . |   (e' ∈ L)
. . . | . . . . . | . |   ⟺
. . . | . . . . . | . |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
```

```
. . . | . . . . . | . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | . |   =
. . . | . . . . . | . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | . 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | . 39. z :
. . . | . . . . . | . |   (tg@0:Id
. . . | . . . . . | . |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                rcv(l; tg@0))) List
. . . | . . . . . | . 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 41. z
. . . | . . . . . | . |   ∈ {p:(tg@0:Id
. . . | . . . . . | . |         ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                      rcv(l; tg@0)))
. . . | . . . . . | . |      | 1of(p) = tg } List
. . . | . . . . . | . 42. tg@0 : Id
. . . | . . . . . | . 43. ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |              rcv(l; tg@0))
. . . | . . . . . | . 44. tg@0 = tg
. . . | . . . . . | . ⊢  if k = rcv(l; tg)→
. . . | . . . . . | . |     if k = rcv(l; tg)→ B else Top fi
. . . | . . . . . | . |  else T fi ⊆r T  by SplitOnConclITE
. . . | . . . . . | . |
. . . | . . . . . | . .....truecase .....
. . . | . . . . . | . 1. x : Id
. . . | . . . . . | . 2. tg : Id
. . . | . . . . . | . 3. k : Knd
. . . | . . . . . | . 4. l : IdLnk
. . . | . . . . . | . 5. T : Type
. . . | . . . . . | . 6. A : Type
. . . | . . . . . | . 7. B : Type
. . . | . . . . . | . 8. f : A→B→(T List)
. . . | . . . . . | . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | . 10. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | . |   ∈ Dsys
. . . | . . . . . | . 11. ∀D:Dsys.
. . . | . . . . . | . |   @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   D
. . . | . . . . . | . |   realizes es.(∀x@0:Id.
. . . | . . . . . | . |                vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . | . |               & (∀e:E.
. . . | . . . . . | . |                   loc(e) = source(l)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   (valtype(e) ⊆r ma-valtype
```

```
. . . | . . . . . | . |                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     kind(e))))
. . . | . . . . . | . |                & (∀e:E.
. . . | . . . . . | . |                   isrcv(e)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   lnk(e) = l
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     kind(e))))
. . . | . . . . . | . |                & (∀e:E.
. . . | . . . . . | . |                   loc(e) = source(l)
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   kind(e) = k
. . . | . . . . . | . |                   ⇒
. . . | . . . . . | . |                   ...)
. . . | . . . . . | . 12. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | . |   ∈ Dsys
. . . | . . . . . | . 13. D : Dsys
. . . | . . . . . | . 14. @source(l): ma-single-sends
. . . | . . . . . | . |   (x : A;
. . . | . . . . . | . |    k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |    k;
. . . | . . . . . | . |    l;
. . . | . . . . . | . |    [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . 15. ∀D':Dsys.
. . . | . . . . . | . |   D ⊆ D'
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀w:World, p:FairFifo.
. . . | . . . . . | . |    PossibleWorld(D';w)
. . . | . . . . . | . |    ⇒
. . . | . . . . . | . |    (∀x@0:Id.
. . . | . . . . . | . |     vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       loc(e) = source(l)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       isrcv(e)
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       lnk(e) = l
. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |        (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |    & (∀e:E.
. . . | . . . . . | . |       loc(e) = source(l)

. . . | . . . . . | . |       ⇒
. . . | . . . . . | . |       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);[<tg
. . . | . . . . . | . |                            ,λs,v. f
. . . | . . . . . | . |                                   (s(x)
. . . | . . . . . | . |                                   ,v)>])))))
. . . | . . . . . | . 16. D' : Dsys
. . . | . . . . . | . 17. D ⊆ D'
. . . | . . . . . | . 18. ∀w:World, p:FairFifo.
. . . | . . . . . | . |   PossibleWorld(D';w)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀x@0:Id.
. . . | . . . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      isrcv(e)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      lnk(e) = l
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      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);[<tg
```

```
. . . | . . . . . | . |                              ,λs,v. f
. . . | . . . . . | . |                                     (s(x)
. . . | . . . . . | . |                                     ,v)>])))
. . . | . . . . . | . 19. w : World
. . . | . . . . . | . 20. ∀p:FairFifo.
. . . | . . . . . | . |   PossibleWorld(D';w)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (∀x@0:Id.
. . . | . . . . . | . |    vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      isrcv(e)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      lnk(e) = l
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |       (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | . |   & (∀e:E.
. . . | . . . . . | . |      loc(e) = source(l)
. . . | . . . . . | . |      ⇒
. . . | . . . . . | . |      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);[<tg
. . . | . . . . . | . |                              ,λs,v. f
. . . | . . . . . | . |                                     (s(x)
. . . | . . . . . | . |                                     ,v)>])))
. . . | . . . . . | . 21. p : FairFifo
. . . | . . . . . | . 22. PossibleWorld(D';w)
. . . | . . . . . | . 23. ∀x@0:Id.
. . . | . . . . . | . |   vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | . 24. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 25. ∀e:E.
. . . | . . . . . | . |   isrcv(e)
. . . | . . . . . | . |   ⇒
```

```
. . . | . . . . . | . |   lnk(e) = l
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | . |    (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 26. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   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);[<tg
. . . | . . . . . | . |                          ,λs,v. f(s(x),v)>]))
. . . | . . . . . | . 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | . 28. ∀e:E.
. . . | . . . . . | . |   loc(e) = source(l)
. . . | . . . . . | . |   ⇒
. . . | . . . . . | . |   kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | . 29. ∀e:E.
. . . | . . . . . | . |   kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | . 30. e : E
. . . | . . . . . | . 31. loc(e) = source(l)
. . . | . . . . . | . 32. kind(e) = k
. . . | . . . . . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | . 34. ∀e':E.
. . . | . . . . . | . |   (e' ∈ L)
. . . | . . . . . | . |   ⟺
. . . | . . . . . | . |   isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | . |   =
. . . | . . . . . | . |   map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | . 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | . 39. z :
. . . | . . . . . | . |   (tg@0:Id
. . . | . . . . . | . |    ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                rcv(l; tg@0))) List
. . . | . . . . . | . 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 41. z
. . . | . . . . . | . |   ∈ {p:(tg@0:Id
. . . | . . . . . | . |         ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |                     rcv(l; tg@0)))
. . . | . . . . . | . |      | 1of(p) = tg } List
. . . | . . . . . | . 42. tg@0 : Id
. . . | . . . . . | . 43. ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | . |             rcv(l; tg@0))
```

```
. . . | . . . . . | . 44. tg@0 = tg
. . . | . . . . . | . 45. k = rcv(l; tg)
. . . | . . . . . | . ⊢  if true₂→ B else Top fi ⊆r T
. . . | . . . . . | . | by Reduce 0
. . . | . . . . . | . |    THEN
. . . | . . . . . | . |    RWO
. . . | . . . . . | . |    Thm* ∀a,b:Knd. a = b ⟺ a = b
. . . | . . . . . | . |    {assert_DASH_eq_DASH_knd}
. . . | . . . . . | . |    -1
. . . | . . . . . | . |    THEN
. . . | . . . . . | . |    SubtypeRelEq
. . . | . . . . . | . |
. . . | . . . . . | . .....equality .....
. . . | . . . . . | . 1. x : Id
. . . | . . . . . | . 2. tg : Id
. . . | . . . . . | . 3. k : Knd
. . . | . . . . . | . 4. l : IdLnk
. . . | . . . . . | . 5. T : Type
. . . | . . . . . | . 6. A : Type
. . . | . . . . . | . 7. B : Type
. . . | . . . . . | . 8. f : A→B→(T List)
. . . | . . . . . | . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . | . 10. @source(l): ma-single-sends
. . . | . . . . . | .     (x : A;
. . . | . . . . . | .      k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | .      k;
. . . | . . . . . | .      l;
. . . | . . . . . | .      [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | .     ∈ Dsys
. . . | . . . . . | . 11. ∀D:Dsys.
. . . | . . . . . | .     @source(l): ma-single-sends
. . . | . . . . . | .     (x : A;

. . . | . . . . . | .      k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | .      k;
. . . | . . . . . | .      l;
. . . | . . . . . | .      [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | .     ⇒
. . . | . . . . . | .     D
. . . | . . . . . | .     realizes es.(∀x@0:Id.
. . . | . . . . . | .                  vartype(source(l);x@0) ⊆r ...)
. . . | . . . . . | .                & (∀e:E.
. . . | . . . . . | .                   loc(e) = source(l)
. . . | . . . . . | .                   ⇒
. . . | . . . . . | .                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | .                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | .                     kind(e))))
. . . | . . . . . | .                & (∀e:E.
. . . | . . . . . | .                   isrcv(e)
. . . | . . . . . | .                   ⇒
. . . | . . . . . | .                   lnk(e) = l
. . . | . . . . . | .                   ⇒
. . . | . . . . . | .                   (valtype(e) ⊆r ma-valtype
. . . | . . . . . | .                    (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | .                     kind(e))))
. . . | . . . . . | .                & (∀e:E.
```

```
. . . | . . . . . | .                    loc(e) = source(l)
. . . | . . . . . | .                    ⇒
. . . | . . . . . | .                    kind(e) = k
. . . | . . . . . | .                    ⇒
. . . | . . . . . | .                    ...)
. . . | . . . . . | . 12. @source(l): ma-single-sends
. . . | . . . . . | .     (x : A;
. . . | . . . . . | .      k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | .      k;
. . . | . . . . . | .      l;
. . . | . . . . . | .      [<tg,λs,v. f(s(x),v)>])
. . . | . . . . . | .     ∈ Dsys
. . . | . . . . . | . 13. D : Dsys
. . . | . . . . . | . 14. @source(l): ma-single-sends
. . . | . . . . . | .     (x : A;
. . . | . . . . . | .      k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | .      k;
. . . | . . . . . | .      l;
. . . | . . . . . | .      [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . | . . . . . | . 15. ∀D':Dsys.
. . . | . . . . . | .     D ⊆ D'
. . . | . . . . . | .     ⇒
. . . | . . . . . | .     (∀w:World, p:FairFifo.
. . . | . . . . . | .      PossibleWorld(D';w)
. . . | . . . . . | .      ⇒
. . . | . . . . . | .      (∀x@0:Id.
. . . | . . . . . | .       vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | .      & (∀e:E.
. . . | . . . . . | .         loc(e) = source(l)
. . . | . . . . . | .         ⇒
. . . | . . . . . | .         (valtype(e) ⊆r ma-valtype
. . . | . . . . . | .          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | .      & (∀e:E.
. . . | . . . . . | .         isrcv(e)
. . . | . . . . . | .         ⇒
. . . | . . . . . | .         lnk(e) = l
. . . | . . . . . | .         ⇒
. . . | . . . . . | .         (valtype(e) ⊆r ma-valtype
. . . | . . . . . | .          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | .      & (∀e:E.
. . . | . . . . . | .         loc(e) = source(l)
. . . | . . . . . | .         ⇒
. . . | . . . . . | .         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);[<tg
. . . | . . . . . | .                              ,λs,v. f
. . . | . . . . . | .                                     (s(x)
. . . | . . . . . | .                                     ,v)>])))))
. . . | . . . . . | . 16. D' : Dsys
. . . | . . . . . | . 17. D ⊆ D'
. . . | . . . . . | . 18. ∀w:World, p:FairFifo.
. . . | . . . . . | .     PossibleWorld(D';w)
. . . | . . . . . | .     ⇒
. . . | . . . . . | .     (∀x@0:Id.
. . . | . . . . . | .      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | .     & (∀e:E.
. . . | . . . . . | .        loc(e) = source(l)
. . . | . . . . . | .        ⇒
. . . | . . . . . | .        (valtype(e) ⊆r ma-valtype
. . . | . . . . . | .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | .     & (∀e:E.
. . . | . . . . . | .        isrcv(e)
. . . | . . . . . | .        ⇒
. . . | . . . . . | .        lnk(e) = l

. . . | . . . . . | .        ⇒
. . . | . . . . . | .        (valtype(e) ⊆r ma-valtype
. . . | . . . . . | .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | .     & (∀e:E.
. . . | . . . . . | .        loc(e) = source(l)
. . . | . . . . . | .        ⇒
. . . | . . . . . | .        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);[<tg
. . . | . . . . . | .                              ,λs,v. f
. . . | . . . . . | .                                     (s(x)
. . . | . . . . . | .                                     ,v)>])))
. . . | . . . . . | . 19. w : World
. . . | . . . . . | . 20. ∀p:FairFifo.
. . . | . . . . . | .     PossibleWorld(D';w)
. . . | . . . . . | .     ⇒
. . . | . . . . . | .     (∀x@0:Id.
. . . | . . . . . | .      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . | .     & (∀e:E.
. . . | . . . . . | .        loc(e) = source(l)
```

```
. . . | . . . . . | .        ⇒
. . . | . . . . . | .        (valtype(e) ⊆r ma-valtype
. . . | . . . . . | .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | .     & (∀e:E.
. . . | . . . . . | .        isrcv(e)
. . . | . . . . . | .        ⇒
. . . | . . . . . | .        lnk(e) = l
. . . | . . . . . | .        ⇒
. . . | . . . . . | .        (valtype(e) ⊆r ma-valtype
. . . | . . . . . | .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . | .     & (∀e:E.
. . . | . . . . . | .        loc(e) = source(l)
. . . | . . . . . | .        ⇒
. . . | . . . . . | .        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);[<tg
. . . | . . . . . | .                              ,λs,v. f
. . . | . . . . . | .                                     (s(x)
. . . | . . . . . | .                                     ,v)>])))
. . . | . . . . . | . 21. p : FairFifo
. . . | . . . . . | . 22. PossibleWorld(D';w)
. . . | . . . . . | . 23. ∀x@0:Id.
. . . | . . . . . | .     vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . | . 24. ∀e:E.
. . . | . . . . . | .     loc(e) = source(l)
. . . | . . . . . | .     ⇒
. . . | . . . . . | .     (valtype(e) ⊆r ma-valtype
. . . | . . . . . | .      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 25. ∀e:E.
. . . | . . . . . | .     isrcv(e)
. . . | . . . . . | .     ⇒
. . . | . . . . . | .     lnk(e) = l
. . . | . . . . . | .     ⇒
. . . | . . . . . | .     (valtype(e) ⊆r ma-valtype
. . . | . . . . . | .      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . | . 26. ∀e:E.
. . . | . . . . . | .     loc(e) = source(l)
. . . | . . . . . | .     ⇒
. . . | . . . . . | .     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);[<tg
. . . | . . . . . | .                           ,λs,v. f(s(x),v)>]))
. . . | . . . . . | . 27. vartype(source(l);x) ⊆r A
. . . | . . . . . | . 28. ∀e:E.
. . . | . . . . . | .     loc(e) = source(l)
. . . | . . . . . | .     ⇒
. . . | . . . . . | .     kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . | . 29. ∀e:E.
. . . | . . . . . | .     kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . | . 30. e : E
. . . | . . . . . | . 31. loc(e) = source(l)
. . . | . . . . . | . 32. kind(e) = k
. . . | . . . . . | . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . | . 34. ∀e':E.
. . . | . . . . . | .     (e' ∈ L)
. . . | . . . . . | .     ⟺
. . . | . . . . . | .     isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . | . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . | . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . | .     =
. . . | . . . . . | .     map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . | . 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . | . 39. z :
. . . | . . . . . | .     (tg@0:Id
. . . | . . . . . | .      ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | .                  rcv(l; tg@0))) List
. . . | . . . . . | . 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . | . 41. z
. . . | . . . . . | .     ∈ {p:(tg@0:Id
. . . | . . . . . | .          ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | .                      rcv(l; tg@0)))
. . . | . . . . . | .        | 1of(p) = tg } List
. . . | . . . . . | . 42. tg@0 : Id
. . . | . . . . . | . 43. ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . | .                rcv(l; tg@0))
. . . | . . . . . | . 44. tg@0 = tg
. . . | . . . . . | . 45. k = rcv(l; tg)
. . . | . . . . . | . ⊢  B = T  by Symmetry THEN BackThruSomeHyp
. . . | . . . . . |
. . . | . . . . .\
. . . | . . . . . 1. x : Id
. . . | . . . . . 2. tg : Id
. . . | . . . . . 3. k : Knd
. . . | . . . . . 4. l : IdLnk
. . . | . . . . . 5. T : Type
. . . | . . . . . 6. A : Type
```

```
. . . | . . . . . 7. B : Type
. . . | . . . . . 8. f : A→B→(T List)
. . . | . . . . . 9. rcv(l; tg) = k ⇒ T = B
. . . | . . . . . 10. @source(l): ma-single-sends(x : A;
. . . | . . . . .                                 k : B ⊕ rcv
. . . | . . . . .                                 (l; tg) : T;
. . . | . . . . .                                 k;
. . . | . . . . .                                 l;
. . . | . . . . .                                 [<tg
. . . | . . . . .                                  ,λs,v. f
. . . | . . . . .                                         (s(x)
. . . | . . . . .                                         ,v)>])
. . . | . . . . .     ∈ Dsys
. . . | . . . . . 11. ∀D:Dsys.
. . . | . . . . .     @source(l): ma-single-sends(x : A;
. . . | . . . . .                                 k : B ⊕ rcv
. . . | . . . . .                                 (l; tg) : T;
. . . | . . . . .                                 k;
. . . | . . . . .                                 l;
. . . | . . . . .                                 [<tg
. . . | . . . . .                                  ,λs,v. f
. . . | . . . . .                                         (s(x)
. . . | . . . . .                                         ,v)>]) ⊆ D
. . . | . . . . .     ⇒
. . . | . . . . .     D
. . . | . . . . .     realizes es.(∀x@0:Id.
. . . | . . . . .                   vartype(source(l);x@0) ⊆r ...)
. . . | . . . . .                  & (∀e:E.
. . . | . . . . .                     loc(e) = source(l)
. . . | . . . . .                     ⇒
. . . | . . . . .                     (valtype(e) ⊆r ma-valtype
. . . | . . . . .                      (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . .                       kind(e))))
. . . | . . . . .                  & (∀e:E.
. . . | . . . . .                     isrcv(e)
. . . | . . . . .                     ⇒
. . . | . . . . .                     lnk(e) = l
. . . | . . . . .                     ⇒
. . . | . . . . .                     (valtype(e) ⊆r ma-valtype
. . . | . . . . .                      (k : B ⊕ rcv(l; tg) : T;
. . . | . . . . .                       kind(e))))
. . . | . . . . .                  & (∀e:E.
. . . | . . . . .                     loc(e) = source(l)
. . . | . . . . .                     ⇒
. . . | . . . . .                     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);[<tg
. . . | . . . . . .                                       ,λs,v.
. . . | . . . . . .                                        f
. . . | . . . . . .                                        (s(x)
. . . | . . . . . .                                        ,v)>])))
. . . | . . . . . . 12. @source(l): ma-single-sends(x : A;
. . . | . . . . . .                                    k : B ⊕ rcv
. . . | . . . . . .                                    (l; tg) : T;
. . . | . . . . . .                                    k;
. . . | . . . . . .                                    l;
. . . | . . . . . .                                    [<tg
. . . | . . . . . .                                     ,λs,v. f
. . . | . . . . . .                                            (s(x)
. . . | . . . . . .                                            ,v)>])
. . . | . . . . . .     ∈ Dsys
. . . | . . . . . . 13. D : Dsys
. . . | . . . . . . 14. @source(l): ma-single-sends(x : A;
. . . | . . . . . .                                    k : B ⊕ rcv
. . . | . . . . . .                                    (l; tg) : T;
. . . | . . . . . .                                    k;
. . . | . . . . . .                                    l;
. . . | . . . . . .                                    [<tg
. . . | . . . . . .                                     ,λs,v. f
. . . | . . . . . .                                            (s(x)
. . . | . . . . . .                                            ,v)>]) ⊆ D
. . . | . . . . . . 15. ∀D':Dsys.
. . . | . . . . . .     D ⊆ D'
. . . | . . . . . .     ⇒
. . . | . . . . . .     (∀w:World, p:FairFifo.
. . . | . . . . . .      PossibleWorld(D';w)
. . . | . . . . . .      ⇒
. . . | . . . . . .      (∀x@0:Id.
. . . | . . . . . .       vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . .      & (∀e:E.
. . . | . . . . . .         loc(e) = source(l)
. . . | . . . . . .         ⇒
. . . | . . . . . .         (valtype(e) ⊆r ma-valtype
. . . | . . . . . .          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . .      & (∀e:E.
. . . | . . . . . .         isrcv(e)
. . . | . . . . . .         ⇒
. . . | . . . . . .         lnk(e) = l
. . . | . . . . . .         ⇒
. . . | . . . . . .         (valtype(e) ⊆r ma-valtype
. . . | . . . . . .          (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . .      & (∀e:E.
. . . | . . . . . .         loc(e) = source(l)
```

```
. . . | . . . . . .          ⇒
. . . | . . . . . .          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);[<tg
. . . | . . . . . .                                ,λs,v. f
. . . | . . . . . .                                       (s(x)
. . . | . . . . . .                                       ,v)>])))))
. . . | . . . . . . 16. D' : Dsys
. . . | . . . . . . 17. D ⊆ D'
. . . | . . . . . . 18. ∀w:World, p:FairFifo.
. . . | . . . . . .     PossibleWorld(D';w)
. . . | . . . . . .     ⇒
. . . | . . . . . .     (∀x@0:Id.
. . . | . . . . . .      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . . .     & (∀e:E.
. . . | . . . . . .        loc(e) = source(l)
. . . | . . . . . .        ⇒
. . . | . . . . . .        (valtype(e) ⊆r ma-valtype
. . . | . . . . . .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . .     & (∀e:E.
. . . | . . . . . .        isrcv(e)
. . . | . . . . . .        ⇒
. . . | . . . . . .        lnk(e) = l
. . . | . . . . . .        ⇒
. . . | . . . . . .        (valtype(e) ⊆r ma-valtype
. . . | . . . . . .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . . .     & (∀e:E.
. . . | . . . . . .        loc(e) = source(l)
. . . | . . . . . .        ⇒
. . . | . . . . . .        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);[<tg
. . . | . . . . . .                              ,λs,v. f(s(x),v)>])))
```

```
. . . | . . . . . 19. w : World
. . . | . . . . . 20. ∀p:FairFifo.
. . . | . . . . .     PossibleWorld(D';w)
. . . | . . . . .     ⇒
. . . | . . . . .     (∀x@0:Id.
. . . | . . . . .      vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . | . . . . .     & (∀e:E.
. . . | . . . . .        loc(e) = source(l)
. . . | . . . . .        ⇒
. . . | . . . . .        (valtype(e) ⊆r ma-valtype
. . . | . . . . .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . .     & (∀e:E.
. . . | . . . . .        isrcv(e)
. . . | . . . . .        ⇒
. . . | . . . . .        lnk(e) = l
. . . | . . . . .        ⇒
. . . | . . . . .        (valtype(e) ⊆r ma-valtype
. . . | . . . . .         (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . | . . . . .     & (∀e:E.
. . . | . . . . .        loc(e) = source(l)
. . . | . . . . .        ⇒
. . . | . . . . .        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);[<tg
. . . | . . . . .                              ,λs,v. f(s(x),v)>])))
. . . | . . . . . 21. p : FairFifo
. . . | . . . . . 22. PossibleWorld(D';w)
. . . | . . . . . 23. ∀x@0:Id.
. . . | . . . . .     vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . | . . . . . 24. ∀e:E.
. . . | . . . . .     loc(e) = source(l)
. . . | . . . . .     ⇒
. . . | . . . . .     (valtype(e) ⊆r ma-valtype
. . . | . . . . .      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . 25. ∀e:E.
. . . | . . . . .     isrcv(e)
. . . | . . . . .     ⇒
. . . | . . . . .     lnk(e) = l
. . . | . . . . .     ⇒
. . . | . . . . .     (valtype(e) ⊆r ma-valtype
. . . | . . . . .      (k : B ⊕ rcv(l; tg) : T; kind(e)))
. . . | . . . . . 26. ∀e:E.
. . . | . . . . .     loc(e) = source(l)
. . . | . . . . .     ⇒
. . . | . . . . .     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);[<tg,λs,v. f(s(x),v)>]))
. . . | . . . . . . 27. vartype(source(l);x) ⊆r A
. . . | . . . . . . 28. ∀e:E.
. . . | . . . . . .     loc(e) = source(l)
. . . | . . . . . .     ⇒
. . . | . . . . . .     kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . | . . . . . . 29. ∀e:E.
. . . | . . . . . .     kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . | . . . . . . 30. e : E
. . . | . . . . . . 31. loc(e) = source(l)
. . . | . . . . . . 32. kind(e) = k
. . . | . . . . . . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . | . . . . . . 34. ∀e':E.
. . . | . . . . . .     (e' ∈ L)
. . . | . . . . . .     ⟺
. . . | . . . . . .     isrcv(e') & lnk(e') = l & sender(e') = e
. . . | . . . . . . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . | . . . . . . 36. map(λe'.<tag(e'),val(e')>;L)
. . . | . . . . . .     =
. . . | . . . . . .     map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . | . . . . . . 38. f((x when e),val(e)) ∈ T List
. . . | . . . . . . 39. z :
. . . | . . . . . .     (tg@0:Id
. . . | . . . . . .      ×ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . | . . . . . .                  rcv(l; tg@0))) List
. . . | . . . . . . 40. z = map(λx@0.<tg,x@0>;f((x when e),val(e)))
. . . | . . . . . . 41. z ∈ (Id×T) List
. . . | . . . . . . ⊢  (z = map(λx@0.<tg,x@0>;f((x when e),val(e))))
. . . | . . . . . .    ∈ Prop  by Auto
. . . |
. . . .\
. . . . .....wf .....
. . . . 1. x : Id
. . . . 2. tg : Id
. . . . 3. k : Knd
. . . . 4. l : IdLnk
. . . . 5. T : Type
. . . . 6. A : Type
. . . . 7. B : Type
. . . . 8. f : A→B→(T List)
. . . . 9. rcv(l; tg) = k ⇒ T = B
. . . . 10. @source(l): ma-single-sends(x : A;
. . . . |                              k : B ⊕ rcv(l; tg) : T;
```

```
. . . . |                                k;
. . . . |                                l;
. . . . |                                [<tg,λs,v. f(s(x),v)>])
. . . . |   ∈ Dsys
. . . . 11. ∀D:Dsys.
. . . . |   @source(l): ma-single-sends(x : A;
. . . . |                                k : B ⊕ rcv(l; tg) : T;
. . . . |                                k;
. . . . |                                l;
. . . . |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . . |   ⇒
. . . . |   D
. . . . |   realizes es.(∀x@0:Id.
. . . . |                 vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . . |               & (∀e:E.
. . . . |                  loc(e) = source(l)
. . . . |                  ⇒
. . . . |                  (valtype(e) ⊆r ma-valtype
. . . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . . |               & (∀e:E.
. . . . |                  isrcv(e)
. . . . |                  ⇒
. . . . |                  lnk(e) = l
. . . . |                  ⇒
. . . . |                  (valtype(e) ⊆r ma-valtype
. . . . |                   (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . . |               & (∀e:E.
. . . . |                  loc(e) = source(l)
. . . . |                  ⇒
. . . . |                  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);[<tg
. . . . |                                        ,λs,v. f(s(x),v)>])))
. . . . 12. @source(l): ma-single-sends(x : A;
. . . . |                                k : B ⊕ rcv(l; tg) : T;
. . . . |                                k;
. . . . |                                l;
. . . . |                                [<tg,λs,v. f(s(x),v)>])
. . . . |   ∈ Dsys
. . . . 13. D : Dsys
. . . . 14. @source(l): ma-single-sends(x : A;
. . . . |                                k : B ⊕ rcv(l; tg) : T;
. . . . |                                k;
. . . . |                                l;
. . . . |                                [<tg,λs,v. f(s(x),v)>]) ⊆ D
```

```
. . . . 15. ∀D':Dsys.
. . . . |   D ⊆ D'
. . . . |   ⇒
. . . . |   (∀w:World, p:FairFifo.
. . . . |    PossibleWorld(D';w)
. . . . |    ⇒
. . . . |    (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . . |    & (∀e:E.
. . . . |       loc(e) = source(l)
. . . . |       ⇒
. . . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . . |                                  kind(e))))
. . . . |    & (∀e:E.
. . . . |       isrcv(e)
. . . . |       ⇒
. . . . |       lnk(e) = l
. . . . |       ⇒
. . . . |       (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . . |                                  kind(e))))
. . . . |    & (∀e:E.
. . . . |       loc(e) = source(l)
. . . . |       ⇒
. . . . |       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);[<tg,λs,v. f(s(x),v)>])))))
. . . . 16. D' : Dsys
. . . . 17. D ⊆ D'
. . . . 18. ∀w:World, p:FairFifo.
. . . . |   PossibleWorld(D';w)
. . . . |   ⇒
. . . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . . |   & (∀e:E.
. . . . |      loc(e) = source(l)
. . . . |      ⇒
. . . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . . |                                  kind(e))))
. . . . |   & (∀e:E.
. . . . |      isrcv(e)
. . . . |      ⇒
. . . . |      lnk(e) = l
. . . . |      ⇒
. . . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . . |                                  kind(e))))
. . . . |   & (∀e:E.
. . . . |      loc(e) = source(l)
. . . . |      ⇒
```

```
. . . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . . 19. w : World
. . . . 20. ∀p:FairFifo.
. . . . |   PossibleWorld(D';w)
. . . . |   ⇒
. . . . |   (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . . |   & (∀e:E.
. . . . |      loc(e) = source(l)
. . . . |      ⇒
. . . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . . |                                 kind(e))))
. . . . |   & (∀e:E.
. . . . |      isrcv(e)
. . . . |      ⇒
. . . . |      lnk(e) = l
. . . . |      ⇒
. . . . |      (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . . |                                 kind(e))))
. . . . |   & (∀e:E.
. . . . |      loc(e) = source(l)
. . . . |      ⇒
. . . . |      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);[<tg,λs,v. f(s(x),v)>])))
. . . . 21. p : FairFifo
. . . . 22. PossibleWorld(D';w)
. . . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . . 24. ∀e:E.
. . . . |   loc(e) = source(l)
. . . . |   ⇒
. . . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . . . |                              ))
. . . . 25. ∀e:E.
. . . . |   isrcv(e)
. . . . |   ⇒
. . . . |   lnk(e) = l
. . . . |   ⇒
. . . . |   (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . . . |                              ))
```

```
. . . . 26. ∀e:E.
. . . . |   loc(e) = source(l)
. . . . |   ⇒
. . . . |   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);[<tg,λs,v. f(s(x),v)>]))
. . . . 27. vartype(source(l);x) ⊆r A
. . . . 28. ∀e:E.
. . . . |   loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)

. . . . 30. e : E
. . . . 31. loc(e) = source(l)
. . . . 32. kind(e) = k
. . . . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . . 34. ∀e':E.
. . . . |   (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . . 36. map(λe'.<tag(e'),val(e')>;L)
. . . . |   =
. . . . |   tagged-list-messages(λz.(z when e);val(e);[<tg
. . . . |                                                ,λs,v. f
. . . . |                                                       (s(x)
. . . . |                                                       ,v)>])
. . . . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . . 38. L1 : {e':E| kind(e') = rcv(l; tg) } List
. . . . ⊢  ((∀e':E.
. . . . |    (e' ∈ L1) ⟺ kind(e') = rcv(l; tg) & sender(e') = e)
. . . . |   & (∀e1,e2:E. e1 before e2 ∈ L1 ⇒ (e1 <loc e2))
. . . . |   & map(λe'.val(e');L1) = f((x when e),val(e)))
. . . . |  ∈ Prop  by MaAuto
. . . . |
. . . . .....wf .....
. . . . 1. x : Id
. . . . 2. tg : Id
. . . . 3. k : Knd
. . . . 4. l : IdLnk
. . . . 5. T : Type
. . . . 6. A : Type
. . . . 7. B : Type
. . . . 8. f : A→B→(T List)
. . . . 9. rcv(l; tg) = k ⇒ T = B
. . . . 10. @source(l): ma-single-sends(x : A;
. . . .                               k : B ⊕ rcv(l; tg) : T;
. . . .                               k;
. . . .                               l;
. . . .                               [<tg,λs,v. f(s(x),v)>])
. . . .     ∈ Dsys
. . . . 11. ∀D:Dsys.
```

```
. . . .     @source(l): ma-single-sends(x : A;
. . . .                                 k : B ⊕ rcv(l; tg) : T;
. . . .                                 k;
. . . .                                 l;
. . . .                                 [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . .     ⇒
. . . .     D
. . . .     realizes es.(∀x@0:Id.
. . . .                   vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . .                  & (∀e:E.
. . . .                     loc(e) = source(l)
. . . .                     ⇒
. . . .                     (valtype(e) ⊆r ma-valtype
. . . .                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . .                  & (∀e:E.
. . . .                     isrcv(e)
. . . .                     ⇒
. . . .                     lnk(e) = l
. . . .                     ⇒
. . . .                     (valtype(e) ⊆r ma-valtype
. . . .                      (k : B ⊕ rcv(l; tg) : T; kind(e))))
. . . .                  & (∀e:E.
. . . .                     loc(e) = source(l)
. . . .                     ⇒
. . . .                     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);[<tg
. . . .                                           ,λs,v. f(s(x),v)>])))
. . . . 12. @source(l): ma-single-sends(x : A;
. . . .                                 k : B ⊕ rcv(l; tg) : T;
. . . .                                 k;
. . . .                                 l;
. . . .                                 [<tg,λs,v. f(s(x),v)>])
. . . .     ∈ Dsys
. . . . 13. D : Dsys
. . . . 14. @source(l): ma-single-sends(x : A;
. . . .                                 k : B ⊕ rcv(l; tg) : T;
. . . .                                 k;
. . . .                                 l;
. . . .                                 [<tg,λs,v. f(s(x),v)>]) ⊆ D
. . . . 15. ∀D':Dsys.
. . . .     D ⊆ D'
. . . .     ⇒
. . . .     (∀w:World, p:FairFifo.
. . . .      PossibleWorld(D';w)
```

```
. . . .      ⇒
. . . .      (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . .      & (∀e:E.
. . . .         loc(e) = source(l)
. . . .         ⇒
. . . .         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . .                                   kind(e))))
. . . .      & (∀e:E.
. . . .         isrcv(e)
. . . .         ⇒
. . . .         lnk(e) = l
. . . .         ⇒
. . . .         (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . .                                   kind(e))))
. . . .      & (∀e:E.
. . . .         loc(e) = source(l)
. . . .         ⇒
. . . .         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);[<tg,λs,v. f(s(x),v)>])))))
. . . . 16. D' : Dsys
. . . . 17. D ⊆ D'
. . . . 18. ∀w:World, p:FairFifo.
. . . .     PossibleWorld(D';w)
. . . .     ⇒
. . . .     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . .     & (∀e:E.
. . . .        loc(e) = source(l)
. . . .        ⇒
. . . .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . .                                  kind(e))))
. . . .     & (∀e:E.
. . . .        isrcv(e)
. . . .        ⇒
. . . .        lnk(e) = l
. . . .        ⇒
. . . .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . .                                  kind(e))))
. . . .     & (∀e:E.
. . . .        loc(e) = source(l)
. . . .        ⇒
. . . .        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);[<tg,λs,v. f(s(x),v)>])))
. . . . 19. w : World
. . . . 20. ∀p:FairFifo.
. . . .     PossibleWorld(D';w)
. . . .     ⇒
. . . .     (∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top)
. . . .     & (∀e:E.
. . . .        loc(e) = source(l)
. . . .        ⇒
. . . .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . .                                  kind(e))))
. . . .     & (∀e:E.
. . . .        isrcv(e)
. . . .        ⇒
. . . .        lnk(e) = l
. . . .        ⇒
. . . .        (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T;
. . . .                                  kind(e))))
. . . .     & (∀e:E.
. . . .        loc(e) = source(l)
. . . .        ⇒
. . . .        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);[<tg,λs,v. f(s(x),v)>])))
. . . . 21. p : FairFifo
. . . . 22. PossibleWorld(D';w)
. . . . 23. ∀x@0:Id. vartype(source(l);x@0) ⊆r x : A(x@0)?Top
. . . . 24. ∀e:E.
. . . .     loc(e) = source(l)
. . . .     ⇒
. . . .     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . . .                               ))
. . . . 25. ∀e:E.
. . . .     isrcv(e)
. . . .     ⇒
. . . .     lnk(e) = l
. . . .     ⇒
. . . .     (valtype(e) ⊆r ma-valtype(k : B ⊕ rcv(l; tg) : T; kind(e)
. . . .                               ))
. . . . 26. ∀e:E.
. . . .     loc(e) = source(l)
. . . .     ⇒
. . . .     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);[<tg,λs,v. f(s(x),v)>]))
. . . . 27. vartype(source(l);x) ⊆r A
. . . . 28. ∀e:E.
. . . .     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . . . 29. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . . . 30. e : E
. . . . 31. loc(e) = source(l)
. . . . 32. kind(e) = k
. . . . 33. L : {e':E| isrcv(e') & lnk(e') = l } List
. . . . 34. ∀e':E.
. . . .     (e' ∈ L) ⟺ isrcv(e') & lnk(e') = l & sender(e') = e
. . . . 35. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . . . 36. map(λe'.<tag(e'),val(e')>;L)
. . . .     =
. . . .     tagged-list-messages(λz.(z when e);val(e);[<tg
. . . .                                                ,λs,v. f
. . . .                                                       (s(x)
. . . .                                                       ,v)>])
. . . . 37. (∀e'∈L.kind(e') = rcv(l; tg))
. . . . 38. {e':E| kind(e') = rcv(l; tg) } List
. . . . 39. e' : {e':E| kind(e') = rcv(l; tg) }
. . . . ⊢  val(e') ∈ T  by Analyze -1 THEN MaAuto
END
```