conditional-send1-rule THM

```
⊢  ∀x,tg:Id, k:Knd, l:IdLnk, T,A,B:Type.
|  (rcv(l; tg) = k ⇒ T = B)
|  ⇒
|  (∀f:(A→B→T), c:(A→B→𝔹).
|   @source(l): ma-single-sends1(A;
|                                B;
|                                T;
|                                x;
|                                k;
|                                l;
|                                tg;
|                                (λa,b. if c(a,b)→ [(f(a,b))]
|                                       else nil fi))
|   ∈ Dsys
|   & (∀D:Dsys.
|      @source(l): ma-single-sends1(A;
|                                   B;
|                                   T;
|                                   x;
|                                   k;
|                                   l;
|                                   tg;
|                                   (λa,b. if c(a,b)→ [(f(a,b))]
|                                          else nil fi)) ⊆ 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
|                     ⇒
|                     (c((x when e),val(e))
|                      ⇒
|                      (∃e':E.
|                       kind(e') = rcv(l; tg)
|                       & sender(e') = e
|                         & (∀e'':E.
|                            kind(e'') = rcv(l; tg)
|                            ⇒
|                            sender(e'') = e ⇒ e'' = e')
|                         & val(e') = f((x when e),val(e))))
|                     & (¬c((x when e),val(e))
|                        ⇒
|                        ¬(∃e':E.
|                          kind(e') = rcv(l; tg) & sender(e') = e)))))
| by Inst Thm: s_DASH_sends_DASH_rule1 []
|    THEN
```

```
|    RepeatFor 7 (ParallelOp -1 THEN Thin -3)
|    THEN
|    RepeatFor 3 (Analyze 0)
|    THEN
|    InstHyp [λa,b. if c(a,b)→ [(f(a,b))] else nil fi] -4
|    THEN
|    Thin -5
|    THEN
|    Analyze -1
|    THEN
|    Analyze 0
|    THEN
|    Try Trivial
|    THEN
|    ParallelOp -2
|    THEN
|    Thin -4
|    THEN
|    ParallelOp -1
|    THEN
|    ParallelOp -1
|    THEN
|    ParallelOp -1
|    THEN
|    Thin -3
|    THEN
|    ParallelOp -1
|    THEN
|    RepeatFor 2 (ParallelOp -1 THEN Thin -3)
|    THEN
|    ParallelOp -1
|    THEN
|    ExRepD
|    THEN
|    BetterSplitAndConcl
|    THEN
|    Try Trivial
|    THEN
|    All Reduce
|    THEN
|    ParallelOp -2
|    THEN
|    Thin -4
|    THEN
|    ParallelLast
|    THEN
|    ParallelLast
|    THEN
|    ThinHypsMentioning [‘ma-single-sends1‘]
|
1. x : Id
2. tg : Id
3. k : Knd
4. l : IdLnk
5. T : Type
```

```
6. A : Type
7. B : Type
8. rcv(l; tg) = k ⇒ T = B
9. f : A→B→T
10. c : A→B→𝔹
11. D : Dsys
12. D' : Dsys
13. D ⊆ D'
14. w : World
15. p : FairFifo
16. PossibleWorld(D';w)
17. vartype(source(l);x) ⊆r A
18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
20. (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))
21. e : E
22. loc(e) = source(l)
23. kind(e) = k
24. ∃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)
|     =
|     if c((x when e),val(e))→ [(f((x when e),val(e)))] else nil fi
⊢  (c((x when e),val(e))
|   ⇒
|   (∃e':E.
|    kind(e') = rcv(l; tg)
|    & sender(e') = e
|      & (∀e'':E.
|         kind(e'') = rcv(l; tg) ⇒ sender(e'') = e ⇒ e'' = e')
|      & val(e') = f((x when e),val(e))))
|  & (¬c((x when e),val(e))
|     ⇒
|     ¬(∃e':E. kind(e') = rcv(l; tg) & sender(e') = e))
| by Analyze 0 THEN Analyze 0 THENA MaAuto THEN SplitOnHypITE -2
|    THEN
|    MaAuto
|
|\
| .....truecase .....
| 1. x : Id
| 2. tg : Id
| 3. k : Knd
| 4. l : IdLnk
| 5. T : Type
| 6. A : Type
| 7. B : Type
| 8. rcv(l; tg) = k ⇒ T = B
| 9. f : A→B→T
| 10. c : A→B→𝔹
| 11. D : Dsys
| 12. D' : Dsys
```

```
| 13. D ⊆ D'
| 14. w : World
| 15. p : FairFifo
| 16. PossibleWorld(D';w)
| 17. vartype(source(l);x) ⊆r A
| 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| 20. vartype(source(l);x) ⊆r A
| 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| 23. e : E
| 24. loc(e) = source(l)
| 25. kind(e) = k
| 26. ∃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)))]
| 27. c((x when e),val(e))
| 28. c((x when e),val(e))
| ⊢  ∃e':E.
| |  kind(e') = rcv(l; tg)
| |  & sender(e') = e
| |    & (∀e'':E.
| |       kind(e'') = rcv(l; tg) ⇒ sender(e'') = e ⇒ e'' = e')
| |    & val(e') = f((x when e),val(e))  by Thin -1 THEN Thin -1
| |
| 1. x : Id
| 2. tg : Id
| 3. k : Knd
| 4. l : IdLnk
| 5. T : Type
| 6. A : Type
| 7. B : Type
| 8. rcv(l; tg) = k ⇒ T = B
| 9. f : A→B→T
| 10. A→B→𝔹
| 11. D : Dsys
| 12. D' : Dsys
| 13. D ⊆ D'
| 14. w : World
| 15. p : FairFifo
| 16. PossibleWorld(D';w)
| 17. vartype(source(l);x) ⊆r A
| 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| 20. vartype(source(l);x) ⊆r A
| 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| 23. e : E
| 24. loc(e) = source(l)
| 25. kind(e) = k
| 26. ∃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)))]
```

```
| ⊢  ∃e':E.
| |  kind(e') = rcv(l; tg)
| |  & sender(e') = e
| |    & (∀e'':E.
| |       kind(e'') = rcv(l; tg) ⇒ sender(e'') = e ⇒ e'' = e')
| |    & val(e') = f((x when e),val(e))
| | by ExRepD THEN Assert (||L|| = 1)
| |
| |\
| | .....assertion .....
| | 1. x : Id
| | 2. tg : Id
| | 3. k : Knd
| | 4. l : IdLnk
| | 5. T : Type
| | 6. A : Type
| | 7. B : Type
| | 8. rcv(l; tg) = k ⇒ T = B
| | 9. f : A→B→T
| | 10. A→B→𝔹
| | 11. D : Dsys
| | 12. D' : Dsys
| | 13. D ⊆ D'
| | 14. w : World
| | 15. p : FairFifo
| | 16. PossibleWorld(D';w)
| | 17. vartype(source(l);x) ⊆r A
| | 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| | 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| | 20. vartype(source(l);x) ⊆r A
| | 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| | 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| | 23. e : E
| | 24. loc(e) = source(l)
| | 25. kind(e) = k
| | 26. L : {e':E| kind(e') = rcv(l; tg) } List
| | 27. ∀e':E. (e' ∈ L) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| | 28. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
| | 29. map(λe'.val(e');L) = [(f((x when e),val(e)))]
| | ⊢  ||L|| = 1
| |   by ApFunToHypEquands ‘Z’ ||Z|| ℤ -1 THEN Reduce -1
| |      THEN
| |      RWO
| |      Thm* ∀f:Top, T:Type, L:T List. ||map(f;L)|| ∼ ||L||
| |      {length_DASH_map}
| |      -1
| |
| .\
| . 1. x : Id
| . 2. tg : Id
| . 3. k : Knd
| . 4. l : IdLnk
| . 5. T : Type
| . 6. A : Type
| . 7. B : Type
```

```
| . 8. rcv(l; tg) = k ⇒ T = B
| . 9. f : A→B→T
| . 10. A→B→𝔹
| . 11. D : Dsys
| . 12. D' : Dsys
| . 13. D ⊆ D'
| . 14. w : World
| . 15. p : FairFifo
| . 16. PossibleWorld(D';w)
| . 17. vartype(source(l);x) ⊆r A
| . 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . 20. vartype(source(l);x) ⊆r A
| . 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . 23. e : E
| . 24. loc(e) = source(l)
| . 25. kind(e) = k
| . 26. L : {e':E| kind(e') = rcv(l; tg) } List
| . 27. ∀e':E. (e' ∈ L) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . 28. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
| . 29. map(λe'.val(e');L) = [(f((x when e),val(e)))]
| . 30. ||L|| = 1
| . ⊢  ∃e':E.
| . |  kind(e') = rcv(l; tg)
| . |  & sender(e') = e
| . |    & (∀e'':E.
| . |       kind(e'') = rcv(l; tg) ⇒ sender(e'') = e ⇒ e'' = e')
| . |    & val(e') = f((x when e),val(e))
| . | by DVar `L' THEN Reduce -1 THEN DVar `v' THEN Reduce -1
| . |    THEN
| . |    Thin -1
| . |    THEN
| . |    Reduce -1
| . |    THEN
| . |    Thin -2
| . |
| . 1. x : Id
| . 2. tg : Id
| . 3. k : Knd
| . 4. l : IdLnk
| . 5. T : Type
| . 6. A : Type
| . 7. B : Type
| . 8. rcv(l; tg) = k ⇒ T = B
| . 9. f : A→B→T
| . 10. A→B→𝔹
| . 11. D : Dsys
| . 12. D' : Dsys
| . 13. D ⊆ D'
| . 14. w : World
| . 15. p : FairFifo
| . 16. PossibleWorld(D';w)
| . 17. vartype(source(l);x) ⊆r A
| . 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
```

```
| . 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . 20. vartype(source(l);x) ⊆r A
| . 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . 23. e : E
| . 24. loc(e) = source(l)
| . 25. kind(e) = k
| . 26. u : {e':E| kind(e') = rcv(l; tg) }
| . 27. ∀e':E. (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . 28. [val(u)] = [(f((x when e),val(e)))]
| . ⊢  ∃e':E.
| . |  kind(e') = rcv(l; tg)
| . |  & sender(e') = e
| . |    & (∀e'':E.
| . |       kind(e'') = rcv(l; tg) ⇒ sender(e'') = e ⇒ e'' = e')
| . |    & val(e') = f((x when e),val(e))  by InstConcl [u]
| . |
| . |\
| . | 1. x : Id
| . | 2. tg : Id
| . | 3. k : Knd
| . | 4. l : IdLnk
| . | 5. T : Type
| . | 6. A : Type
| . | 7. B : Type
| . | 8. rcv(l; tg) = k ⇒ T = B
| . | 9. f : A→B→T
| . | 10. A→B→𝔹
| . | 11. D : Dsys
| . | 12. D' : Dsys
| . | 13. D ⊆ D'
| . | 14. w : World
| . | 15. p : FairFifo
| . | 16. PossibleWorld(D';w)
| . | 17. vartype(source(l);x) ⊆r A
| . | 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 20. vartype(source(l);x) ⊆r A
| . | 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 23. e : E
| . | 24. loc(e) = source(l)
| . | 25. kind(e) = k
| . | 26. u : {e':E| kind(e') = rcv(l; tg) }
| . | 27. ∀e':E. (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . | 28. [val(u)] = [(f((x when e),val(e)))]
| . | ⊢  kind(u) = rcv(l; tg)  by DVar ‘u’
| . |
| . |\
| . | 1. x : Id
| . | 2. tg : Id
| . | 3. k : Knd
| . | 4. l : IdLnk
| . | 5. T : Type
| . | 6. A : Type
```

```
| . | 7. B : Type
| . | 8. rcv(l; tg) = k ⇒ T = B
| . | 9. f : A→B→T
| . | 10. A→B→𝔹
| . | 11. D : Dsys
| . | 12. D' : Dsys
| . | 13. D ⊆ D'
| . | 14. w : World
| . | 15. p : FairFifo
| . | 16. PossibleWorld(D';w)
| . | 17. vartype(source(l);x) ⊆r A
| . | 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 20. vartype(source(l);x) ⊆r A
| . | 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 23. e : E
| . | 24. loc(e) = source(l)
| . | 25. kind(e) = k
| . | 26. u : {e':E| kind(e') = rcv(l; tg) }
| . | 27. ∀e':E. (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . | 28. [val(u)] = [(f((x when e),val(e)))]
| . | ⊢  sender(u) = e
| . |   by InstHyp [u] -2 THEN Analyze -2 THEN ExRepD
| . |      THEN
| . |      RWO Thm* ∀a,x:T. (x ∈ [a]) ⟺ x = a    {member_singleton}
| . |      0
| . |
| . |\
| . | 1. x : Id
| . | 2. tg : Id
| . | 3. k : Knd
| . | 4. l : IdLnk
| . | 5. T : Type
| . | 6. A : Type
| . | 7. B : Type
| . | 8. rcv(l; tg) = k ⇒ T = B
| . | 9. f : A→B→T
| . | 10. A→B→𝔹
| . | 11. D : Dsys
| . | 12. D' : Dsys
| . | 13. D ⊆ D'
| . | 14. w : World
| . | 15. p : FairFifo
| . | 16. PossibleWorld(D';w)
| . | 17. vartype(source(l);x) ⊆r A
| . | 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 20. vartype(source(l);x) ⊆r A
| . | 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 23. e : E
| . | 24. loc(e) = source(l)
| . | 25. kind(e) = k
| . | 26. u : {e':E| kind(e') = rcv(l; tg) }
```

```
| . | 27. ∀e':E. (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . | 28. [val(u)] = [(f((x when e),val(e)))]
| . | 29. e'' : E
| . | 30. kind(e'') = rcv(l; tg)
| . | 31. sender(e'') = e
| . | ⊢  e'' = u  by RWO "member_singleton<" 0 THEN BackThruSomeHyp
| . |
| . |\
| . | 1. x : Id
| . | 2. tg : Id
| . | 3. k : Knd
| . | 4. l : IdLnk
| . | 5. T : Type
| . | 6. A : Type
| . | 7. B : Type
| . | 8. rcv(l; tg) = k ⇒ T = B
| . | 9. f : A→B→T
| . | 10. A→B→𝔹
| . | 11. D : Dsys
| . | 12. D' : Dsys
| . | 13. D ⊆ D'
| . | 14. w : World
| . | 15. p : FairFifo
| . | 16. PossibleWorld(D';w)
| . | 17. vartype(source(l);x) ⊆r A
| . | 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 20. vartype(source(l);x) ⊆r A
| . | 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 23. e : E
| . | 24. loc(e) = source(l)
| . | 25. kind(e) = k
| . | 26. u : {e':E| kind(e') = rcv(l; tg) }
| . | 27. ∀e':E. (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . | 28. [val(u)] = [(f((x when e),val(e)))]
| . | 29. e'' : E
| . | 30. 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. rcv(l; tg) = k ⇒ T = B
| . | 9. f : A→B→T
| . | 10. A→B→𝔹
| . | 11. D : Dsys
| . | 12. D' : Dsys
| . | 13. D ⊆ D'
| . | 14. w : World
| . | 15. p : FairFifo
| . | 16. PossibleWorld(D';w)
| . | 17. vartype(source(l);x) ⊆r A
| . | 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 20. vartype(source(l);x) ⊆r A
| . | 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 23. e : E
| . | 24. loc(e) = source(l)
| . | 25. kind(e) = k
| . | 26. u : {e':E| kind(e') = rcv(l; tg) }
| . | 27. ∀e':E. (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . | 28. [val(u)] = [(f((x when e),val(e)))]
| . | ⊢  val(u) = f((x when e),val(e))
| . | | by Subst (val(u) ~ hd([val(u)])) 0
| . | |
| . | |\
| . | | .....equality .....
| . | | 1. x : Id
| . | | 2. tg : Id
| . | | 3. k : Knd
| . | | 4. l : IdLnk
| . | | 5. T : Type
| . | | 6. A : Type
| . | | 7. B : Type
| . | | 8. rcv(l; tg) = k ⇒ T = B
| . | | 9. f : A→B→T
| . | | 10. A→B→𝔹
| . | | 11. D : Dsys
| . | | 12. D' : Dsys
| . | | 13. D ⊆ D'
| . | | 14. w : World
| . | | 15. p : FairFifo
| . | | 16. PossibleWorld(D';w)
| . | | 17. vartype(source(l);x) ⊆r A
| . | | 18. ∀e:E.
| . | |     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | | 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | | 20. vartype(source(l);x) ⊆r A
| . | | 21. ∀e:E.
| . | |     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | | 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
```

```
| . | | 23. e : E
| . | | 24. loc(e) = source(l)
| . | | 25. kind(e) = k
| . | | 26. u : {e':E| kind(e') = rcv(l; tg) }
| . | | 27. ∀e':E.
| . | |     (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . | | 28. [val(u)] = [(f((x when e),val(e)))]
| . | | ⊢  val(u) ∼ hd([val(u)])  by Reduce 0
| . | |
| . | .\
| . | . 1. x : Id
| . | . 2. tg : Id
| . | . 3. k : Knd
| . | . 4. l : IdLnk
| . | . 5. T : Type
| . | . 6. A : Type
| . | . 7. B : Type
| . | . 8. rcv(l; tg) = k ⇒ T = B
| . | . 9. f : A→B→T
| . | . 10. A→B→𝔹
| . | . 11. D : Dsys
| . | . 12. D' : Dsys
| . | . 13. D ⊆ D'
| . | . 14. w : World
| . | . 15. p : FairFifo
| . | . 16. PossibleWorld(D';w)
| . | . 17. vartype(source(l);x) ⊆r A
| . | . 18. ∀e:E.
| . | .     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | . 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | . 20. vartype(source(l);x) ⊆r A
| . | . 21. ∀e:E.
| . | .     loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | . 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | . 23. e : E
| . | . 24. loc(e) = source(l)
| . | . 25. kind(e) = k
| . | . 26. u : {e':E| kind(e') = rcv(l; tg) }
| . | . 27. ∀e':E.
| . | .     (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . | . 28. [val(u)] = [(f((x when e),val(e)))]
| . | . ⊢  hd([val(u)]) = f((x when e),val(e))
| . | .   by (StrongHypSubst -1 0 THEN Reduce 0 THEN MaAuto
| . | .       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. rcv(l; tg) = k ⇒ T = B
| . | 9. f : A→B→T
| . | 10. A→B→𝔹
| . | 11. D : Dsys
| . | 12. D' : Dsys
| . | 13. D ⊆ D'
| . | 14. w : World
| . | 15. p : FairFifo
| . | 16. PossibleWorld(D';w)
| . | 17. vartype(source(l);x) ⊆r A
| . | 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 20. vartype(source(l);x) ⊆r A
| . | 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 23. e : E
| . | 24. loc(e) = source(l)
| . | 25. kind(e) = k
| . | 26. u : {e':E| kind(e') = rcv(l; tg) }
| . | 27. ∀e':E. (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . | 28. [val(u)] = [(f((x when e),val(e)))]
| . | 29. e' : E
| . | 30. 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. rcv(l; tg) = k ⇒ T = B
| . | 9. f : A→B→T
| . | 10. A→B→𝔹
| . | 11. D : Dsys
| . | 12. D' : Dsys
| . | 13. D ⊆ D'
| . | 14. w : World
| . | 15. p : FairFifo
| . | 16. PossibleWorld(D';w)
```

```
| . | 17. vartype(source(l);x) ⊆r A
| . | 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 20. vartype(source(l);x) ⊆r A
| . | 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 23. e : E
| . | 24. loc(e) = source(l)
| . | 25. kind(e) = k
| . | 26. u : {e':E| kind(e') = rcv(l; tg) }
| . | 27. ∀e':E. (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . | 28. [val(u)] = [(f((x when e),val(e)))]
| . | 29. e' : E
| . | 30. kind(e') = rcv(l; tg)
| . | 31. e'' : E
| . | 32. 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
| . |
| . |\
| . | .....subterm .....
| . | 1. x : Id
| . | 2. tg : Id
| . | 3. k : Knd
| . | 4. l : IdLnk
| . | 5. T : Type
| . | 6. A : Type
| . | 7. B : Type
| . | 8. rcv(l; tg) = k ⇒ T = B
| . | 9. f : A→B→T
| . | 10. A→B→𝔹
| . | 11. D : Dsys
| . | 12. D' : Dsys
| . | 13. D ⊆ D'
| . | 14. w : World
| . | 15. p : FairFifo
| . | 16. PossibleWorld(D';w)
| . | 17. vartype(source(l);x) ⊆r A
| . | 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 20. vartype(source(l);x) ⊆r A
| . | 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 23. e : E
```

```
| . | 24. loc(e) = source(l)
| . | 25. kind(e) = k
| . | 26. u : {e':E| kind(e') = rcv(l; tg) }
| . | 27. ∀e':E. (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . | 28. [val(u)] = [(f((x when e),val(e)))]
| . | 29. e' : E
| . | 30. kind(e') = rcv(l; tg)
| . | ⊢  val(e') ∈ T  by MaAuto
| . |
| . |\
| . | .....subterm .....
| . | 1. x : Id
| . | 2. tg : Id
| . | 3. k : Knd
| . | 4. l : IdLnk
| . | 5. T : Type
| . | 6. A : Type
| . | 7. B : Type
| . | 8. rcv(l; tg) = k ⇒ T = B
| . | 9. f : A→B→T
| . | 10. A→B→𝔹
| . | 11. D : Dsys
| . | 12. D' : Dsys
| . | 13. D ⊆ D'
| . | 14. w : World
| . | 15. p : FairFifo
| . | 16. PossibleWorld(D';w)
| . | 17. vartype(source(l);x) ⊆r A
| . | 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 20. vartype(source(l);x) ⊆r A
| . | 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . | 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . | 23. e : E
| . | 24. loc(e) = source(l)
| . | 25. kind(e) = k
| . | 26. u : {e':E| kind(e') = rcv(l; tg) }
| . | 27. ∀e':E. (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . | 28. [val(u)] = [(f((x when e),val(e)))]
| . | 29. e' : E
| . | 30. kind(e') = rcv(l; tg)
| . | ⊢  (x when e) ∈ A  by MaAuto
| . |
| . .\
| . . .....subterm .....
| . . 1. x : Id
| . . 2. tg : Id
| . . 3. k : Knd
| . . 4. l : IdLnk
| . . 5. T : Type
| . . 6. A : Type
| . . 7. B : Type
| . . 8. rcv(l; tg) = k ⇒ T = B
| . . 9. f : A→B→T
| . . 10. A→B→𝔹
```

```
| . . 11. D : Dsys
| . . 12. D' : Dsys
| . . 13. D ⊆ D'
| . . 14. w : World
| . . 15. p : FairFifo
| . . 16. PossibleWorld(D';w)
| . . 17. vartype(source(l);x) ⊆r A
| . . 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . . 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . . 20. vartype(source(l);x) ⊆r A
| . . 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
| . . 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
| . . 23. e : E
| . . 24. loc(e) = source(l)
| . . 25. kind(e) = k
| . . 26. u : {e':E| kind(e') = rcv(l; tg) }
| . . 27. ∀e':E. (e' ∈ [u]) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
| . . 28. [val(u)] = [(f((x when e),val(e)))]
| . . 29. e' : E
| . . 30. kind(e') = rcv(l; tg)
| . . ⊢  val(e) ∈ B  by MaAuto
|
.\
. .....falsecase .....
. 1. x : Id
. 2. tg : Id
. 3. k : Knd
. 4. l : IdLnk
. 5. T : Type
. 6. A : Type
. 7. B : Type
. 8. rcv(l; tg) = k ⇒ T = B

. 9. A→B→T
. 10. c : A→B→𝔹
. 11. D : Dsys
. 12. D' : Dsys
. 13. D ⊆ D'
. 14. w : World
. 15. p : FairFifo
. 16. PossibleWorld(D';w)
. 17. vartype(source(l);x) ⊆r A
. 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. 20. vartype(source(l);x) ⊆r A
. 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. 23. e : E
. 24. loc(e) = source(l)
. 25. kind(e) = k
. 26. ∃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) = nil
. 27. ¬c((x when e),val(e))
. 28. ¬c((x when e),val(e))
```

```
. ⊢  ¬(∃e':E. kind(e') = rcv(l; tg) & sender(e') = e)
. | by Thin -1 THEN Thin -1 THEN ExRepD THEN Subst (L ∼ nil) -3
. |
. |\
. | .....equality .....
. | 1. x : Id
. | 2. tg : Id
. | 3. k : Knd
. | 4. l : IdLnk
. | 5. T : Type
. | 6. A : Type
. | 7. B : Type
. | 8. rcv(l; tg) = k ⇒ T = B
. | 9. A→B→T
. | 10. A→B→𝔹
. | 11. D : Dsys
. | 12. D' : Dsys
. | 13. D ⊆ D'
. | 14. w : World
. | 15. p : FairFifo
. | 16. PossibleWorld(D';w)
. | 17. vartype(source(l);x) ⊆r A
. | 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. | 20. vartype(source(l);x) ⊆r A
. | 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. | 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. | 23. e : E
. | 24. loc(e) = source(l)
. | 25. kind(e) = k
. | 26. L : {e':E| kind(e') = rcv(l; tg) } List
. | 27. ∀e':E. (e' ∈ L) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
. | 28. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. | 29. map(λe'.val(e');L) = nil
. | ⊢  L ∼ nil
. |   by ApFunToHypEquands `Z' ||Z|| ℤ -1 THEN Reduce -1
. |      THEN
. |      RWO
. |      Thm* ∀f:Top, T:Type, L:T List. ||map(f;L)|| ∼ ||L||
. |      {length_DASH_map}
. |      -1
. |      THEN
. |      DVar `L'
. |      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. rcv(l; tg) = k ⇒ T = B
```

```
. . 9. A⟶B⟶T
. . 10. A⟶B⟶𝔹
. . 11. D : Dsys
. . 12. D' : Dsys
. . 13. D ⊆ D'
. . 14. w : World
. . 15. p : FairFifo
. . 16. PossibleWorld(D';w)
. . 17. vartype(source(l);x) ⊆r A
. . 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . 20. vartype(source(l);x) ⊆r A
. . 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . 23. e : E
. . 24. loc(e) = source(l)
. . 25. kind(e) = k
. . 26. L : {e':E| kind(e') = rcv(l; tg) } List
. . 27. ∀e':E. (e' ∈ nil) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
. . 28. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . 29. map(λe'.val(e');L) = nil
. . ⊢  ¬(∃e':E. kind(e') = rcv(l; tg) & sender(e') = e)

. . | by Analyze 0 THENA MaAuto THEN ExRepD
. . |
. . 1. x : Id
. . 2. tg : Id
. . 3. k : Knd
. . 4. l : IdLnk
. . 5. T : Type
. . 6. A : Type
. . 7. B : Type
. . 8. rcv(l; tg) = k ⇒ T = B
. . 9. A⟶B⟶T
. . 10. A⟶B⟶𝔹
. . 11. D : Dsys
. . 12. D' : Dsys
. . 13. D ⊆ D'
. . 14. w : World
. . 15. p : FairFifo
. . 16. PossibleWorld(D';w)
. . 17. vartype(source(l);x) ⊆r A
. . 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . 20. vartype(source(l);x) ⊆r A
. . 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . 23. e : E
. . 24. loc(e) = source(l)
. . 25. kind(e) = k
. . 26. L : {e':E| kind(e') = rcv(l; tg) } List
. . 27. ∀e':E. (e' ∈ nil) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
. . 28. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . 29. map(λe'.val(e');L) = nil
. . 30. e' : E
. . 31. kind(e') = rcv(l; tg)
```

```
. . 32. sender(e') = e
. . ⊢  False  by InstHyp [e'] -6 THEN Analyze -1
. . |
. . 1. x : Id
. . 2. tg : Id
. . 3. k : Knd
. . 4. l : IdLnk
. . 5. T : Type
. . 6. A : Type
. . 7. B : Type
. . 8. rcv(l; tg) = k ⇒ T = B
. . 9. A→B→T
. . 10. A→B→𝔹
. . 11. D : Dsys
. . 12. D' : Dsys
. . 13. D ⊆ D'
. . 14. w : World
. . 15. p : FairFifo
. . 16. PossibleWorld(D';w)
. . 17. vartype(source(l);x) ⊆r A
. . 18. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . 19. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . 20. vartype(source(l);x) ⊆r A
. . 21. ∀e:E. loc(e) = source(l) ⇒ kind(e) = k ⇒ (valtype(e) ⊆r B)
. . 22. ∀e:E. kind(e) = rcv(l; tg) ⇒ (valtype(e) ⊆r T)
. . 23. e : E
. . 24. loc(e) = source(l)
. . 25. kind(e) = k
. . 26. L : {e':E| kind(e') = rcv(l; tg) } List
. . 27. ∀e':E. (e' ∈ nil) ⟺ kind(e') = rcv(l; tg) & sender(e') = e
. . 28. ∀e1,e2:E. e1 before e2 ∈ L ⇒ (e1 <loc e2)
. . 29. map(λe'.val(e');L) = nil
. . 30. e' : E
. . 31. kind(e') = rcv(l; tg)
. . 32. sender(e') = e
. . 33. (e' ∈ nil) ⇒ kind(e') = rcv(l; tg) & sender(e') = e
. . 34. (e' ∈ nil)
. . ⊢  False  by Analyze -1 THEN ExRepD THEN All Reduce
END
```