```
world-event-system  THM

⊢  ∀the_w:World.
|  FairFifo
|  ⇒
|  ESAxioms{i:l}
|          (E;
|           (λi,x. vartype(i;x));
|           the_w.M;
|           (λe.loc(e));
|           (λe.kind(e));
|           (λe.val(e));
|           (λx,e. (x when e));
|           (λx,e. (x after e));
|           (λl,e. sends(l;e));
|           (λe.sender(e));
|           (λe.index(e));
|           (λe.first(e));
|           (λe.pred(e));
|           (λe,e'. e <c e'))
| by RepeatFor 2 (Analyze 0) THEN Unfold `ESAxioms` 0 THEN Reduce 0
|    THEN
|    BetterSplitAndConcl
|
|\
| 1. the_w : World
| 2. FairFifo
| ⊢  ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | by Auto THEN Assert (e <c e' ⟺ e <loc e')
| |
| |\
| | .....assertion .....
| | 1. the_w : World
| | 2. FairFifo
| | 3. e : E
| | 4. e' : E
| | 5. loc(e) = loc(e')
| | ⊢  e <c e' ⟺ e <loc e'
| |   by Inst
| |      Thm* ∀the_w:World, e,e':E.
| |           FairFifo
| |           ⇒
| |           (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| |      {w_DASH_locl_DASH_iff}
| |      [the_w;e;e']
| |      THEN
| |      ThinTrivial
| |      THEN
| |      BackThruSomeHyp
| |
| .\
| . 1. the_w : World
| . 2. FairFifo
| . 3. e : E
| . 4. e' : E
```

```
| . 5. loc(e) = loc(e')
| . 6. e <c e' ⟺ e <loc e'
| . ⊢  e <c e' ∨ e = e' ∨ e' <c e
| . | by RW (SweepDnC (HypC -1)) 0 THEN Assert (e' <c e ⟺ e' <loc e)
| . |
| . |\
| . | .....assertion .....
| . | 1. the_w : World
| . | 2. FairFifo
| . | 3. e : E
| . | 4. e' : E
| . | 5. loc(e) = loc(e')
| . | 6. e <c e' ⟺ e <loc e'
| . | ⊢  e' <c e ⟺ e' <loc e
| . |   by Inst
| . |      Thm* ∀the_w:World, e,e':E.
| . |           FairFifo
| . |           ⇒
| . |           (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| . |      {w_DASH_locl_DASH_iff}
| . |      [the_w;e';e]
| . |      THEN
| . |      ThinTrivial
| . |      THEN
| . |      BackThruSomeHyp
| . |
| . .\
| . . 1. the_w : World
| . . 2. FairFifo
| . . 3. e : E
| . . 4. e' : E
| . . 5. loc(e) = loc(e')
| . . 6. e <c e' ⟺ e <loc e'
| . . 7. e' <c e ⟺ e' <loc e
| . . ⊢  e <loc e' ∨ e = e' ∨ e' <c e  by RW (SweepDnC (HypC -1)) 0
| . . |
| . . 1. the_w : World
| . . 2. FairFifo
| . . 3. e : E
| . . 4. e' : E
| . . 5. loc(e) = loc(e')
| . . 6. e <c e' ⟺ e <loc e'
| . . 7. e' <c e ⟺ e' <loc e
| . . ⊢  e <loc e' ∨ e = e' ∨ e' <loc e
| . . | by Unfold `w-locl` 0 THEN Decide (time(e)<time(e'))
| . . |    THENL
| . . |    [OrLeft;OrRight]
| . . |    THEN
| . . |    Decide (time(e')<time(e))
| . . |    THENL
| . . |    [OrRight;OrLeft]
| . . |
| . . 1. the_w : World
| . . 2. FairFifo
| . . 3. e : E
```

```
| . . 4. e' : E
| . . 5. loc(e) = loc(e')
| . . 6. e <c e' ⇒ e <loc e'
| . . 7. e <c e' ⇐ e <loc e'
| . . 8. e' <c e ⇒ e' <loc e
| . . 9. e' <c e ⇐ e' <loc e
| . . 10. ¬time(e)<time(e')
| . . 11. ¬time(e')<time(e)
| . . ⊢  e = e'  by Assert (time(e) = time(e'))
| . . |
| . . 1. the_w : World
| . . 2. FairFifo
| . . 3. e : E
| . . 4. e' : E
| . . 5. loc(e) = loc(e')
| . . 6. e <c e' ⇒ e <loc e'
| . . 7. e <c e' ⇐ e <loc e'
| . . 8. e' <c e ⇒ e' <loc e
| . . 9. e' <c e ⇐ e' <loc e
| . . 10. ¬time(e)<time(e')
| . . 11. ¬time(e')<time(e)
| . . 12. time(e) = time(e')
| . . ⊢  e = e'
| . .   by All (Unfolds [`w-time`;`w-loc`]) THEN Analyze 4
| . .      THEN
| . .      Analyze 4
| . .      THEN
| . .      Analyze 3
| . .      THEN
| . .      Analyze 3
| . .      THEN
| . .      All Reduce
| . .      THEN
| . .      Analyze
| . .      THEN
| . .      Reduce 0
| . .      THEN
| . .      Analyze
|
|\
| 1. the_w : World
| 2. FairFifo
| 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| ⊢  ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | by Assert (∀e:E. first(e) ⟺ (∀e':E. ¬e' <loc e))
| |
| |\
| | .....assertion .....
| | 1. the_w : World
| | 2. FairFifo
| | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | ⊢  ∀e:E. first(e) ⟺ (∀e':E. ¬e' <loc e)  by Analyze 0
| | |
| | |\
| | | 1. the_w : World
```

```
| | | 2. FairFifo
| | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | 4. e : E
| | | 5. first(e)
| | | 6. e' : E
| | | ⊢  ¬e' <loc e
| | | | by (Analyze 0) THEN (Unfold `w-locl` -1)
| | | |    THEN
| | | |    (RWO
| | | |     Thm* ∀the_w:World, e:E.
| | | |          first(e)
| | | |          ⟺
| | | |          (∀t':ℕ. t'<time(e) ⇒ isnull(a(loc(e);t')))
| | | |     {assert_DASH_w_DASH_first}
| | | |     -3)
| | | |    THEN
| | | |    (InstHyp [time(e')] -3)
| | | |
| | | 1. the_w : World
| | | 2. FairFifo
| | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | 4. e : E
| | | 5. ∀t':ℕ. t'<time(e) ⇒ isnull(a(loc(e);t'))
| | | 6. e' : E
| | | 7. loc(e') = loc(e)
| | | 8. time(e')<time(e)
| | | 9. isnull(a(loc(e);time(e')))
| | | ⊢  False
| | |   by ((RevHypSubst -3 -1)
| | |       THEN
| | |       (Inst
| | |        Thm* ∀the_w:World, e:E. ¬isnull(act(e))
| | |        {w_DASH_act_DASH_not_DASH_null}
| | |        [the_w;e'])
| | |       THEN
| | |       (Analyze -1)
| | |       THEN
| | |       (Unfold `w-act` 0))
| | |      THEN
| | |      NthHypSq -1
| | |      THEN
| | |      RepeatFor 3 Analyze
| | |      THEN
| | |      Try (Unfolds [`w-loc`;`w-time`] 0)
| | |      THEN
| | |      Try Trivial
| | |
| | .\
| | . 1. the_w : World
| | . 2. FairFifo
| | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . 4. e : E
| | . 5. ∀e':E. ¬e' <loc e
| | . ⊢  first(e)
| | . | by RWO
```

```
| | . |    Thm* ∀the_w:World, e:E.
| | . |         first(e)
| | . |         ⟺
| | . |         (∀t':ℕ. t'<time(e) ⇒ isnull(a(loc(e);t')))
| | . |    {assert_DASH_w_DASH_first}
| | . |    0
| | . |    THEN
| | . |    SupposeNot
| | . |    THEN
| | . |    InstHyp [<loc(e),t'>] -4
| | . |
| | . |\
| | . | .....wf .....
| | . | 1. the_w : World
| | . | 2. FairFifo
| | . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . | 4. e : E
| | . | 5. ∀e':E. ¬e' <loc e
| | . | 6. t' : ℕ
| | . | 7. t'<time(e)
| | . | 8. ¬isnull(a(loc(e);t'))
| | . | ⊢  <loc(e),t'> ∈ E
| | . |   by Unfold `w-E` 0 THEN Analyze THEN Reduce 0
| | . |
| | . .\
| | . . 1. the_w : World
| | . . 2. FairFifo
| | . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . 4. e : E
| | . . 5. ∀e':E. ¬e' <loc e
| | . . 6. t' : ℕ
| | . . 7. t'<time(e)
| | . . 8. ¬isnull(a(loc(e);t'))
| | . . 9. ¬<loc(e),t'> <loc e
| | . . ⊢  isnull(a(loc(e);t'))
| | . .   by Analyze -1 THEN Unfold `w-locl` 0
| | . .      THEN
| | . .      All (Unfolds [`w-time`;`w-loc`])
| | . .      THEN
| | . .      Analyze 4
| | . .      THEN
| | . .      Analyze 4
| | . .      THEN
| | . .      All Reduce
| |
| .\
| . 1. the_w : World
| . 2. FairFifo
| . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . 4. ∀e:E. first(e) ⟺ (∀e':E. ¬e' <loc e)
| . ⊢  ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | by ParallelOp -1 THEN ParallelOp -1 THEN ParallelOp -1
| . |
| . |\
| . | 1. the_w : World
```

```
| . | 2. FairFifo
| . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | 4. ∀e:E. first(e) ⟺ (∀e':E. ¬e' <loc e)
| . | 5. e : E
| . | 6. first(e)
| . | 7. ∀e':E. ¬e' <loc e
| . | 8. e' : E
| . | 9. ¬e' <loc e
| . | ⊢  loc(e') = loc(e) ⇒ ¬e' <c e
| . |   by RWO
| . |      Thm* ∀the_w:World, e,e':E.
| . |           FairFifo
| . |           ⇒
| . |           (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| . |      {w_DASH_locl_DASH_iff}
| . |      -1
| . |      THEN
| . |      ParallelOp -2
| . |
| . .\
| . . 1. the_w : World
| . . 2. FairFifo
| . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . . 4. ∀e:E. first(e) ⟺ (∀e':E. ¬e' <loc e)
| . . 5. e : E
| . . 6. ∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e
| . . 7. e' : E
| . . 8. loc(e') = loc(e) ⇒ ¬e' <c e
| . . ⊢  ¬e' <loc e
| . .   by ((RWO
| . .         Thm* ∀the_w:World, e,e':E.
| . .              FairFifo
| . .              ⇒
| . .              (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| . .         {w_DASH_locl_DASH_iff}
| . .         0)
| . .        THEN
| . .        (Analyze 0))
| . .      THEN
| . .      ThinTrivial
|
|\
| 1. the_w : World
| 2. FairFifo
| 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| ⊢  ∀e:E.
| |  ¬first(e)
| |  ⇒
| |  loc(pred(e)) = loc(e) & pred(e) <c e
| |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | by Assert
| |    (∀e:E.
| |     ¬first(e)
| |     ⇒
```

```
| |     pred(e) <loc e & (∀e':E. ¬(pred(e) <loc e' & e' <loc e)))
| |
| |\
| | .....assertion .....
| | 1. the_w : World
| | 2. FairFifo
| | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | ⊢  ∀e:E.
| | |  ¬first(e)
| | |  ⇒
| | |  pred(e) <loc e & (∀e':E. ¬(pred(e) <loc e' & e' <loc e))
| | | by Assert
| | |    (∀t:ℕ, i:Id.
| | |     ¬first(<i,t>)
| | |     ⇒
| | |     pred(<i,t>) <loc <i,t>
| | |     & (∀e':E. ¬(pred(<i,t>) <loc e' & e' <loc <i,t>)))
| | |
| | |\
| | | .....assertion .....
| | | 1. the_w : World
| | | 2. FairFifo
| | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | ⊢  ∀t:ℕ, i:Id.
| | | |  ¬first(<i,t>)
| | | |  ⇒
| | | |  pred(<i,t>) <loc <i,t>
| | | |  & (∀e':E. ¬(pred(<i,t>) <loc e' & e' <loc <i,t>))
| | | | by InductionOnNat THEN RecUnfold `w-first` 0
| | | |    THEN
| | | |    RecUnfold `w-pred` 0
| | | |    THEN
| | | |    Unfolds [`w-loc`;`w-time`] 0
| | | |    THEN
| | | |    Reduce 0
| | | |    THEN
| | | |    Try
| | | |    (Complete ((Analyze 0) THEN (Analyze 0) THEN (Analyze -1)))
| | | |    THEN
| | | |    SplitOnConclITE
| | | |    THENL
| | | |    [Auto;Id]
| | | |
| | | .....falsecase .....
| | | 1. the_w : World
| | | 2. FairFifo
| | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | 5. t : ℤ
| | | 6. 0<t
| | | 7. ∀i:Id.
| | | |  ¬first(<i,t-1>)
| | | |  ⇒
```

```
| | | |  pred(<i,t-1>) <loc <i,t-1>
| | | |  & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | 8. ¬t = 0
| | | ⊢  ∀i:Id.
| | | |  ¬if isnull(a(i;t-1))→ first(<i,t-1>) else false2 fi
| | | |  ⇒
| | | |  if isnull(a(i;t-1))→ pred(<i,t-1>) else <i,t-1> fi <loc <i,t>
| | | |  & (∀e':E.
| | | |     ¬(if isnull(a(i;t-1))→ pred(<i,t-1>)
| | | |       else <i,t-1> fi <loc e'
| | | |       & e' <loc <i,t>))  by (Analyze 0) THEN SplitOnConclITE
| | | |
| | | |\
| | | | .....truecase .....
| | | | 1. the_w : World
| | | | 2. FairFifo
| | | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | 4. ∀e:E. first(e) ⇐⇒ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | 5. t : ℤ
| | | | 6. 0<t
| | | | 7. ∀i:Id.
| | | | |  ¬first(<i,t-1>)
| | | | |  ⇒
| | | | |  pred(<i,t-1>) <loc <i,t-1>
| | | | |  & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | | 8. ¬t = 0
| | | | 9. i : Id
| | | | 10. isnull(a(i;t-1))
| | | | ⊢  ¬first(<i,t-1>)
| | | | |  ⇒
| | | | |  pred(<i,t-1>) <loc <i,t>
| | | | |  & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t>))
| | | | | by Assert (first(<i,t-1>) ∈ 𝔹)
| | | | |
| | | | |\
| | | | | .....assertion .....
| | | | | 1. the_w : World
| | | | | 2. FairFifo
| | | | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | | 4. ∀e:E. first(e) ⇐⇒ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | | 5. t : ℤ
| | | | | 6. 0<t
| | | | | 7. ∀i:Id.
| | | | |    ¬first(<i,t-1>)
| | | | |    ⇒
| | | | |    pred(<i,t-1>) <loc <i,t-1>
| | | | |    & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | | | 8. ¬t = 0
| | | | | 9. i : Id
| | | | | 10. isnull(a(i;t-1))
| | | | | ⊢  first(<i,t-1>) ∈ 𝔹
| | | | |   by BackThru
| | | | |      Thm* ∀the_w:World, i:Id, t:ℕ. first(<i,t>) ∈ 𝔹
| | | | |      {w_DASH_first_DASH_aux}
| | | | |
```

```
| | | | .\
| | | | . 1. the_w : World
| | | | . 2. FairFifo
| | | | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . 5. t : ℤ
| | | | . 6. 0<t
| | | | . 7. ∀i:Id.
| | | | . |  ¬first(<i,t-1>)
| | | | . |  ⇒
| | | | . |  pred(<i,t-1>) <loc <i,t-1>
| | | | . |  & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | | . 8. ¬t = 0
| | | | . 9. i : Id
| | | | . 10. isnull(a(i;t-1))
| | | | . 11. first(<i,t-1>) ∈ 𝔹
| | | | . ⊢  ¬first(<i,t-1>)
| | | | . |  ⇒
| | | | . |  pred(<i,t-1>) <loc <i,t>
| | | | . |  & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t>))
| | | | . | by Analyze 0 THEN InstHyp [i] 7
| | | | . |
| | | | . |\
| | | | . | 1. the_w : World
| | | | . | 2. FairFifo
| | | | . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . | 5. t : ℤ
| | | | . | 6. 0<t
| | | | . | 7. ∀i:Id.
| | | | . | |  ¬first(<i,t-1>)
| | | | . | |  ⇒
| | | | . | |  pred(<i,t-1>) <loc <i,t-1>
| | | | . | |  & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | | . | 8. ¬t = 0
| | | | . | 9. i : Id
| | | | . | 10. isnull(a(i;t-1))
| | | | . | 11. first(<i,t-1>) ∈ 𝔹
| | | | . | 12. ¬first(<i,t-1>)
| | | | . | 13. pred(<i,t-1>) <loc <i,t-1>
| | | | . | 14. ∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>)
| | | | . | ⊢  pred(<i,t-1>) <loc <i,t>
| | | | . | | by All (Unfold `w-locl`) THEN All Reduce
| | | | . | |
| | | | . | 1. the_w : World
| | | | . | 2. FairFifo
| | | | . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . | 5. t : ℤ
| | | | . | 6. 0<t
| | | | . | 7. ∀i:Id.
| | | | . | |  ¬first(<i,t-1>)
| | | | . | |  ⇒
| | | | . | |  loc(pred(<i,t-1>)) = i & time(pred(<i,t-1>))<t-1
| | | | . | |  & (∀e':E.
```

```
| | | | . | |     ¬(loc(pred(<i,t-1>)) = loc(e')
| | | | . | |       & time(pred(<i,t-1>))<time(e')
| | | | . | |       & loc(e') = i
| | | | . | |       & time(e')<t-1))
| | | | . | 8. ¬t = 0
| | | | . | 9. i : Id
| | | | . | 10. isnull(a(i;t-1))
| | | | . | 11. first(<i,t-1>) ∈ 𝔹
| | | | . | 12. ¬first(<i,t-1>)
| | | | . | 13. loc(pred(<i,t-1>)) = i & time(pred(<i,t-1>))<t-1
| | | | . | 14. ∀e':E.
| | | | . | |   ¬(loc(pred(<i,t-1>)) = loc(e')
| | | | . | |     & time(pred(<i,t-1>))<time(e')
| | | | . | |     & loc(e') = i
| | | | . | |     & time(e')<t-1)
| | | | . | ⊢  loc(pred(<i,t-1>)) = i & time(pred(<i,t-1>))<t
| | | | . | | by Analyze 0
| | | | . | |
| | | | . | |\
| | | | . | | 1. the_w : World
| | | | . | | 2. FairFifo
| | | | . | | 3. ∀e,e':E.
| | | | . | |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . | | 4. ∀e:E.
| | | | . | |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . | | 5. t : ℤ
| | | | . | | 6. 0<t
| | | | . | | 7. ∀i:Id.
| | | | . | |    ¬first(<i,t-1>)
| | | | . | |    ⇒
| | | | . | |    loc(pred(<i,t-1>)) = i & time(pred(<i,t-1>))<t-1
| | | | . | |    & (∀e':E.
| | | | . | |       ¬(loc(pred(<i,t-1>)) = loc(e')
| | | | . | |         & time(pred(<i,t-1>))<time(e')
| | | | . | |         & loc(e') = i
| | | | . | |         & time(e')<t-1))
| | | | . | | 8. ¬t = 0
| | | | . | | 9. i : Id
| | | | . | | 10. isnull(a(i;t-1))
| | | | . | | 11. first(<i,t-1>) ∈ 𝔹
| | | | . | | 12. ¬first(<i,t-1>)
| | | | . | | 13. loc(pred(<i,t-1>)) = i & time(pred(<i,t-1>))<t-1
| | | | . | | 14. ∀e':E.
| | | | . | |     ¬(loc(pred(<i,t-1>)) = loc(e')
| | | | . | |       & time(pred(<i,t-1>))<time(e')
| | | | . | |       & loc(e') = i
| | | | . | |       & time(e')<t-1)
| | | | . | | ⊢  loc(pred(<i,t-1>)) = i  by Auto
| | | | . | |
| | | | . | .\
| | | | . | . 1. the_w : World
| | | | . | . 2. FairFifo
| | | | . | . 3. ∀e,e':E.
| | | | . | . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . | . 4. ∀e:E.
```

```
| | | | . | . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . | . 5. t : ℤ
| | | | . | . 6. 0<t
| | | | . | . 7. ∀i:Id.
| | | | . | . |  ¬first(<i,t-1>)
| | | | . | . |  ⇒
| | | | . | . |  loc(pred(<i,t-1>)) = i & time(pred(<i,t-1>))<t-1
| | | | . | . |  & (∀e':E.
| | | | . | . |     ¬(loc(pred(<i,t-1>)) = loc(e')
| | | | . | . |       & time(pred(<i,t-1>))<time(e')
| | | | . | . |       & loc(e') = i
| | | | . | . |       & time(e')<t-1))
| | | | . | . 8. ¬t = 0
| | | | . | . 9. i : Id
| | | | . | . 10. isnull(a(i;t-1))
| | | | . | . 11. first(<i,t-1>) ∈ 𝔹
| | | | . | . 12. ¬first(<i,t-1>)
| | | | . | . 13. loc(pred(<i,t-1>)) = i & time(pred(<i,t-1>))<t-1
| | | | . | . 14. ∀e':E.
| | | | . | . |   ¬(loc(pred(<i,t-1>)) = loc(e')
| | | | . | . |     & time(pred(<i,t-1>))<time(e')
| | | | . | . |     & loc(e') = i
| | | | . | . |     & time(e')<t-1)
| | | | . | . ⊢  time(pred(<i,t-1>))<t
| | | | . | . | by Analyze -2 THEN MoveToConcl -2
| | | | . | . |    THEN
| | | | . | . |    GenConcl (pred(<i,t-1>) = p)
| | | | . | . |    THEN
| | | | . | . |    Try (Complete Auto)
| | | | . | . |
| | | | . | . .....wf .....
| | | | . | . 1. the_w : World
| | | | . | . 2. FairFifo
| | | | . | . 3. ∀e,e':E.
| | | | . | .    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . | . 4. ∀e:E.
| | | | . | .    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . | . 5. t : ℤ
| | | | . | . 6. 0<t
| | | | . | . 7. ∀i:Id.
| | | | . | .    ¬first(<i,t-1>)
| | | | . | .    ⇒
| | | | . | .    loc(pred(<i,t-1>)) = i & time(pred(<i,t-1>))<t-1
| | | | . | .    & (∀e':E.
| | | | . | .       ¬(loc(pred(<i,t-1>)) = loc(e')
| | | | . | .         & time(pred(<i,t-1>))<time(e')
| | | | . | .         & loc(e') = i
| | | | . | .         & time(e')<t-1))
| | | | . | . 8. ¬t = 0
| | | | . | . 9. i : Id
| | | | . | . 10. isnull(a(i;t-1))
| | | | . | . 11. first(<i,t-1>) ∈ 𝔹
| | | | . | . 12. ¬first(<i,t-1>)
| | | | . | . 13. loc(pred(<i,t-1>)) = i
| | | | . | . 14. ∀e':E.
```

```
| | | | . | .     ¬(loc(pred(<i,t-1>)) = loc(e')
| | | | . | .       & time(pred(<i,t-1>))<time(e')
| | | | . | .       & loc(e') = i
| | | | . | .       & time(e')<t-1)
| | | | . | . ⊢  pred(<i,t-1>) ∈ E
| | | | . | .   by BackThru
| | | | . | .      Thm* ∀the_w:World, t:ℕ, i:Id.
| | | | . | .           ¬first(<i,t>) ⇒ pred(<i,t>) ∈ E
| | | | . | .      {w_DASH_pred_DASH_aux}
| | | | . |
| | | | . .\
| | | | . . 1. the_w : World
| | | | . . 2. FairFifo
| | | | . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . . 5. t : ℤ
| | | | . . 6. 0<t
| | | | . . 7. ∀i:Id.
| | | | . . |  ¬first(<i,t-1>)
| | | | . . |  ⇒
| | | | . . |  pred(<i,t-1>) <loc <i,t-1>
| | | | . . |  & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | | . . 8. ¬t = 0
| | | | . . 9. i : Id
| | | | . . 10. isnull(a(i;t-1))
| | | | . . 11. first(<i,t-1>) ∈ 𝔹
| | | | . . 12. ¬first(<i,t-1>)
| | | | . . 13. pred(<i,t-1>) <loc <i,t-1>
| | | | . . 14. ∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>)
| | | | . . 15. e' : E
| | | | . . ⊢  ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t>)
| | | | . . | by InstHyp [e'] -2 THEN MoveToConcl -1
| | | | . . |    THEN
| | | | . . |    GenConcl (pred(<i,t-1>) = e'')
| | | | . . |
| | | | . . |\
| | | | . . | .....wf .....
| | | | . . | 1. the_w : World
| | | | . . | 2. FairFifo
| | | | . . | 3. ∀e,e':E.
| | | | . . |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . . | 4. ∀e:E.
| | | | . . |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . . | 5. t : ℤ
| | | | . . | 6. 0<t
| | | | . . | 7. ∀i:Id.
| | | | . . |    ¬first(<i,t-1>)
| | | | . . |    ⇒
| | | | . . |    pred(<i,t-1>) <loc <i,t-1>
| | | | . . |    & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | | . . | 8. ¬t = 0
| | | | . . | 9. i : Id
| | | | . . | 10. isnull(a(i;t-1))
| | | | . . | 11. first(<i,t-1>) ∈ 𝔹
| | | | . . | 12. ¬first(<i,t-1>)
```

```
| | | | . . | 13. pred(<i,t-1>) <loc <i,t-1>
| | | | . . | 14. ∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>)
| | | | . . | 15. E
| | | | . . | ⊢  pred(<i,t-1>) ∈ E
| | | | . . |   by BackThru
| | | | . . |      Thm* ∀the_w:World, t:ℕ, i:Id.
| | | | . . |           ¬first(<i,t>) ⇒ pred(<i,t>) ∈ E
| | | | . . |      {w_DASH_pred_DASH_aux}
| | | | . . |
| | | | . . |\
| | | | . . | .....wf .....
| | | | . . | 1. the_w : World
| | | | . . | 2. FairFifo
| | | | . . | 3. ∀e,e':E.
| | | | . . |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . . | 4. ∀e:E.
| | | | . . |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . . | 5. t : ℤ
| | | | . . | 6. 0<t
| | | | . . | 7. ∀i:Id.
| | | | . . |    ¬first(<i,t-1>)
| | | | . . |    ⇒
| | | | . . |    pred(<i,t-1>) <loc <i,t-1>
| | | | . . |    & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | | . . | 8. ¬t = 0
| | | | . . | 9. i : Id
| | | | . . | 10. isnull(a(i;t-1))
| | | | . . | 11. first(<i,t-1>) ∈ 𝔹
| | | | . . | 12. ¬first(<i,t-1>)
| | | | . . | 13. pred(<i,t-1>) <loc <i,t-1>
| | | | . . | 14. ∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>)
| | | | . . | 15. E
| | | | . . | ⊢  E ∈ Type  by Auto
| | | | . . |
| | | | . . .\
| | | | . . . 1. the_w : World
| | | | . . . 2. FairFifo
| | | | . . . 3. ∀e,e':E.
| | | | . . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . . . 4. ∀e:E.
| | | | . . . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . . . 5. t : ℤ

| | | | . . . 6. 0<t
| | | | . . . 7. ∀i:Id.
| | | | . . . |  ¬first(<i,t-1>)
| | | | . . . |  ⇒
| | | | . . . |  pred(<i,t-1>) <loc <i,t-1>
| | | | . . . |  & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | | . . . 8. ¬t = 0
| | | | . . . 9. i : Id
| | | | . . . 10. isnull(a(i;t-1))
| | | | . . . 11. first(<i,t-1>) ∈ 𝔹
| | | | . . . 12. ¬first(<i,t-1>)
| | | | . . . 13. pred(<i,t-1>) <loc <i,t-1>
| | | | . . . 14. ∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>)
```

```
| | | | . . . 15. e' : E
| | | | . . . 16. e'' : E
| | | | . . . 17. pred(<i,t-1>) = e''
| | | | . . . ⊢  ¬(e'' <loc e' & e' <loc <i,t-1>)
| | | | . . . |  ⇒
| | | | . . . |  ¬(e'' <loc e' & e' <loc <i,t>)
| | | | . . . | by Unfold `w-locl` 0 THEN Reduce 0
| | | | . . . |
| | | | . . . 1. the_w : World
| | | | . . . 2. FairFifo
| | | | . . . 3. ∀e,e':E.
| | | | . . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . . . 4. ∀e:E.
| | | | . . . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . . . 5. t : ℤ
| | | | . . . 6. 0<t
| | | | . . . 7. ∀i:Id.
| | | | . . . |  ¬first(<i,t-1>)
| | | | . . . |  ⇒
| | | | . . . |  pred(<i,t-1>) <loc <i,t-1>
| | | | . . . |  & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | | . . . 8. ¬t = 0
| | | | . . . 9. i : Id
| | | | . . . 10. isnull(a(i;t-1))
| | | | . . . 11. first(<i,t-1>) ∈ 𝔹
| | | | . . . 12. ¬first(<i,t-1>)
| | | | . . . 13. pred(<i,t-1>) <loc <i,t-1>
| | | | . . . 14. ∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>)
| | | | . . . 15. e' : E
| | | | . . . 16. e'' : E
| | | | . . . 17. pred(<i,t-1>) = e''
| | | | . . . 18. ¬(loc(e'') = loc(e') & time(e'')<time(e')
| | | | . . . |     & loc(e') = i
| | | | . . . |     & time(e')<t-1)
| | | | . . . ⊢  ¬(loc(e'') = loc(e') & time(e'')<time(e')
| | | | . . . |    & loc(e') = i
| | | | . . . |    & time(e')<t)  by ParallelOp -1
| | | | . . . |
| | | | . . . 1. the_w : World
| | | | . . . 2. FairFifo
| | | | . . . 3. ∀e,e':E.
| | | | . . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . . . 4. ∀e:E.
| | | | . . . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . . . 5. t : ℤ
| | | | . . . 6. 0<t
| | | | . . . 7. ∀i:Id.
| | | | . . . |  ¬first(<i,t-1>)
| | | | . . . |  ⇒
| | | | . . . |  pred(<i,t-1>) <loc <i,t-1>
| | | | . . . |  & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | | . . . 8. ¬t = 0
| | | | . . . 9. i : Id
| | | | . . . 10. isnull(a(i;t-1))
| | | | . . . 11. first(<i,t-1>) ∈ 𝔹
```

```
| | | | . . . 12. ¬first(<i,t-1>)
| | | | . . . 13. pred(<i,t-1>) <loc <i,t-1>
| | | | . . . 14. ∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>)
| | | | . . . 15. e' : E
| | | | . . . 16. e'' : E
| | | | . . . 17. pred(<i,t-1>) = e''
| | | | . . . 18. loc(e'') = loc(e')
| | | | . . . 19. time(e'')<time(e')
| | | | . . . 20. loc(e') = i
| | | | . . . 21. time(e')<t
| | | | . . . ⊢  time(e')<t-1  by Decide (time(e') = t-1)
| | | | . . . |
| | | | . . . 1. the_w : World
| | | | . . . 2. FairFifo
| | | | . . . 3. ∀e,e':E.
| | | | . . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . . . 4. ∀e:E.
| | | | . . . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . . . 5. t : ℤ
| | | | . . . 6. 0<t
| | | | . . . 7. ∀i:Id.
| | | | . . . |  ¬first(<i,t-1>)
| | | | . . . |  ⇒
| | | | . . . |  pred(<i,t-1>) <loc <i,t-1>
| | | | . . . |  & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | | . . . 8. ¬t = 0
| | | | . . . 9. i : Id
| | | | . . . 10. isnull(a(i;t-1))
| | | | . . . 11. first(<i,t-1>) ∈ 𝔹
| | | | . . . 12. ¬first(<i,t-1>)
| | | | . . . 13. pred(<i,t-1>) <loc <i,t-1>
| | | | . . . 14. ∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>)

| | | | . . . 15. e' : E
| | | | . . . 16. e'' : E
| | | | . . . 17. pred(<i,t-1>) = e''
| | | | . . . 18. loc(e'') = loc(e')
| | | | . . . 19. time(e'')<time(e')
| | | | . . . 20. loc(e') = i
| | | | . . . 21. time(e')<t
| | | | . . . 22. time(e') = t-1
| | | | . . . ⊢  time(e')<t-1  by Assert False
| | | | . . . |
| | | | . . . .....assertion .....
| | | | . . . 1. the_w : World
| | | | . . . 2. FairFifo
| | | | . . . 3. ∀e,e':E.
| | | | . . .    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | . . . 4. ∀e:E.
| | | | . . .    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | . . . 5. t : ℤ
| | | | . . . 6. 0<t
| | | | . . . 7. ∀i:Id.
| | | | . . .    ¬first(<i,t-1>)
| | | | . . .    ⇒
| | | | . . .    pred(<i,t-1>) <loc <i,t-1>
```

```
| | | | . . .    & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | | . . . 8. ¬t = 0
| | | | . . . 9. i : Id
| | | | . . . 10. isnull(a(i;t-1))
| | | | . . . 11. first(<i,t-1>) ∈ 𝔹
| | | | . . . 12. ¬first(<i,t-1>)
| | | | . . . 13. pred(<i,t-1>) <loc <i,t-1>
| | | | . . . 14. ∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>)
| | | | . . . 15. e' : E
| | | | . . . 16. e'' : E
| | | | . . . 17. pred(<i,t-1>) = e''
| | | | . . . 18. loc(e'') = loc(e')
| | | | . . . 19. time(e'')<time(e')
| | | | . . . 20. loc(e') = i
| | | | . . . 21. time(e')<t
| | | | . . . 22. time(e') = t-1
| | | | . . . ⊢  False
| | | | . . .   by Analyze 15 THEN Analyze 16
| | | | . . .      THEN
| | | | . . .      All (Unfolds [`w-loc`;`w-time`])
| | | | . . .      THEN
| | | | . . .      All Reduce
| | | | . . .      THEN
| | | | . . .      HypSubst' -1 0
| | | | . . .      THEN
| | | | . . .      HypSubst -3 0
| | | |
| | | .\
| | | . .....falsecase .....
| | | . 1. the_w : World
| | | . 2. FairFifo
| | | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | . 5. t : ℤ
| | | . 6. 0<t
| | | . 7. ∀i:Id.
| | | .    ¬first(<i,t-1>)
| | | .    ⇒
| | | .    pred(<i,t-1>) <loc <i,t-1>
| | | .    & (∀e':E. ¬(pred(<i,t-1>) <loc e' & e' <loc <i,t-1>))
| | | . 8. ¬t = 0
| | | . 9. i : Id
| | | . 10. ¬isnull(a(i;t-1))
| | | . ⊢  ¬false_2
| | | .    ⇒
| | | .    <i,t-1> <loc <i,t>
| | | .    & (∀e':E. ¬(<i,t-1> <loc e' & e' <loc <i,t>))
| | | .   by Unfold `w-locl` 0 THEN Reduce 0 THEN Analyze 0
| | |
| | .\
| | . 1. the_w : World
| | . 2. FairFifo
| | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . 5. ∀t:ℕ, i:Id.
```

```
| | .    ¬first(<i,t>)
| | .    ⇒
| | .    pred(<i,t>) <loc <i,t>
| | .    & (∀e':E. ¬(pred(<i,t>) <loc e' & e' <loc <i,t>))
| | . ⊢  ∀e:E.
| | .    ¬first(e)
| | .    ⇒
| | .    pred(e) <loc e & (∀e':E. ¬(pred(e) <loc e' & e' <loc e))
| | .   by RepeatFor 2 (Analyze 0) THEN InstHyp [time(e);loc(e)] 5
| | .      THEN
| | .      All
| | .      (RWO
| | .       Thm* ∀the_w:World, e:E. <loc(e),time(e)> ~ e
| | .       {w_DASH_loc_DASH_time})
| |
| .\
| . 1. the_w : World
| . 2. FairFifo
| . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . 5. ∀e:E.
| . |  ¬first(e)
| . |  ⇒
| . |  pred(e) <loc e & (∀e':E. ¬(pred(e) <loc e' & e' <loc e))
| . ⊢  ∀e:E.
| . |  ¬first(e)
| . |  ⇒
| . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | by (RepeatFor 3 (ParallelOp -1)
| . |     THEN
| . |     RWO
| . |     Thm* ∀the_w:World, e,e':E.
| . |          FairFifo
| . |          ⇒
| . |          (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| . |     {w_DASH_locl_DASH_iff}
| . |     -1
| . |     THEN
| . |     Analyze 0)
| . |    THEN
| . |    (InstHyp [e'] -4)
| . |    THEN
| . |    (Analyze -1)
| . |
| . 1. the_w : World
| . 2. FairFifo
| . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . 5. ∀e:E.
| . |  ¬first(e)
| . |  ⇒
| . |  pred(e) <loc e & (∀e':E. ¬(pred(e) <loc e' & e' <loc e))
| . 6. e : E
| . 7. ¬first(e)
```

```
| . 8. ∀e':E.
| . |  ¬(loc(pred(e)) = loc(e') & pred(e) <c e'
| . |    & loc(e') = loc(e)
| . |    & e' <c e)
| . 9. e' : E
| . 10. loc(e') = loc(e)
| . 11. pred(e) <c e'
| . 12. e' <c e
| . ⊢  loc(pred(e)) = loc(e')  by InstHyp [e] 5
| . |
| . 1. the_w : World
| . 2. FairFifo
| . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . 5. ∀e:E.
| .    ¬first(e)
| .    ⇒
| .    pred(e) <loc e & (∀e':E. ¬(pred(e) <loc e' & e' <loc e))
| . 6. e : E
| . 7. ¬first(e)
| . 8. ∀e':E.
| .    ¬(loc(pred(e)) = loc(e') & pred(e) <c e'
| .      & loc(e') = loc(e)
| .      & e' <c e)
| . 9. e' : E
| . 10. loc(e') = loc(e)
| . 11. pred(e) <c e'
| . 12. e' <c e
| . 13. pred(e) <loc e
| . 14. ∀e':E. ¬(pred(e) <loc e' & e' <loc e)
| . ⊢  loc(pred(e)) = loc(e')
| .   by RWO
| .      Thm* ∀the_w:World, e,e':E.
| .           FairFifo
| .           ⇒
| .           (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| .      {w_DASH_locl_DASH_iff}
| .      -2
|
|\
| 1. the_w : World
| 2. FairFifo
| 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| 5. ∀e:E.
| |  ¬first(e)
| |  ⇒
| |  loc(pred(e)) = loc(e) & pred(e) <c e
| |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| ⊢  ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | by Assert
| |    (∀t:ℕ, i:Id.
| |     ¬first(<i,t>)
| |     ⇒
| |     (∀x:Id. (x when <i,t>) = (x after pred(<i,t>))))
```

```
| |
| |\
| | .....assertion .....
| | 1. the_w : World
| | 2. FairFifo
| | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | 5. ∀e:E.
| | |  ¬first(e)
| | |  ⇒
| | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | ⊢  ∀t:ℕ, i:Id.
| | |  ¬first(<i,t>)
| | |  ⇒
| | |  (∀x:Id. (x when <i,t>) = (x after pred(<i,t>)))
| | | by (InductionOnNat THEN RecUnfold `w-first` 0
| | |     THEN
| | |     RecUnfold `w-pred` 0
| | |     THEN
| | |     Reduce 0
| | |     THEN
| | |     Try (Complete (Auto THEN Analyze -2))
| | |     THEN
| | |     SplitOnConclITE
| | |     THENL
| | |     [Auto;Id]
| | |     THEN
| | |     Analyze 0)
| | |    THEN
| | |    SplitOnConclITE
| | |
| | |\
| | | .....truecase .....
| | | 1. the_w : World
| | | 2. FairFifo
| | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | 5. ∀e:E.
| | | |  ¬first(e)
| | | |  ⇒
| | | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | 6. t : ℤ
| | | 7. 0<t
| | | 8. ∀i:Id.
| | | |  ¬first(<i,t-1>)
| | | |  ⇒
| | | |  (∀x:Id. (x when <i,t-1>) = (x after pred(<i,t-1>)))
| | | 9. ¬t = 0
| | | 10. i : Id
| | | 11. isnull(a(i;t-1))
| | | ⊢  ¬first(<i,t-1>)
| | | |  ⇒
| | | |  (∀x@0:Id. (x@0 when <i,t>) = (x@0 after pred(<i,t-1>)))
```

```
| | | | | by (AssertBY (first(<i,t-1>) ∈ 𝔹)
| | | | |     (BackThru
| | | | |      Thm* ∀the_w:World, i:Id, t:ℕ. first(<i,t>) ∈ 𝔹
| | | | |      {w_DASH_first_DASH_aux})
| | | | |     THEN
| | | | |     Analyze 0)
| | | | |    THEN
| | | | |    (InstHyp [i] 8)
| | | | |
| | | | 1. the_w : World
| | | | 2. FairFifo
| | | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | 5. ∀e:E.
| | | | |  ¬first(e)
| | | | |  ⇒
| | | | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | | | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | | 6. t : ℤ
| | | | 7. 0<t
| | | | 8. ∀i:Id.
| | | | |  ¬first(<i,t-1>)
| | | | |  ⇒
| | | | |  (∀x:Id. (x when <i,t-1>) = (x after pred(<i,t-1>)))
| | | | 9. ¬t = 0
| | | | 10. i : Id
| | | | 11. isnull(a(i;t-1))
| | | | 12. first(<i,t-1>) ∈ 𝔹
| | | | 13. ¬first(<i,t-1>)
| | | | 14. ∀x:Id. (x when <i,t-1>) = (x after pred(<i,t-1>))
| | | | ⊢  ∀x@0:Id. (x@0 when <i,t>) = (x@0 after pred(<i,t-1>))
| | | | | by ParallelOp -1
| | | | |    THEN
| | | | |    Assert ((x@0 when <i,t>) = (x@0 when <i,t-1>))
| | | | |
| | | | .....assertion .....
| | | | 1. the_w : World
| | | | 2. FairFifo
| | | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | 5. ∀e:E.
| | | | |  ¬first(e)
| | | | |  ⇒
| | | | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | | | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | | 6. t : ℤ
| | | | 7. 0<t
| | | | 8. ∀i:Id.
| | | | |  ¬first(<i,t-1>)
| | | | |  ⇒
| | | | |  (∀x:Id. (x when <i,t-1>) = (x after pred(<i,t-1>)))
| | | | 9. ¬t = 0
| | | | 10. i : Id
| | | | 11. isnull(a(i;t-1))
| | | | 12. first(<i,t-1>) ∈ 𝔹
```

```
| | | 13. ¬first(<i,t-1>)
| | | 14. ∀x:Id. (x when <i,t-1>) = (x after pred(<i,t-1>))
| | | 15. x@0 : Id
| | | 16. (x@0 when <i,t-1>) = (x@0 after pred(<i,t-1>))
| | | ⊢  (x@0 when <i,t>) = (x@0 when <i,t-1>)
| | | | by Unfold ‘w-when‘ 0 THEN Reduce 0
| | | |
| | | 1. the_w : World
| | | 2. FairFifo
| | | 3. ∀e,e’:E. loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
| | | 4. ∀e:E. first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
| | | 5. ∀e:E.
| | |    ¬first(e)
| | |    ⇒
| | |    loc(pred(e)) = loc(e) & pred(e) <c e
| | |    & (∀e’:E. loc(e’) = loc(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
| | | 6. t : ℤ
| | | 7. 0<t
| | | 8. ∀i:Id.
| | |    ¬first(<i,t-1>)
| | |    ⇒
| | |    (∀x:Id. (x when <i,t-1>) = (x after pred(<i,t-1>)))
| | | 9. ¬t = 0
| | | 10. i : Id
| | | 11. isnull(a(i;t-1))
| | | 12. first(<i,t-1>) ∈ 𝔹
| | | 13. ¬first(<i,t-1>)
| | | 14. ∀x:Id. (x when <i,t-1>) = (x after pred(<i,t-1>))
| | | 15. x@0 : Id
| | | 16. (x@0 when <i,t-1>) = (x@0 after pred(<i,t-1>))
| | | ⊢  s(i;t).x@0 = s(i;t-1).x@0
| | |   by Unfold ‘fair-fifo‘ 2 THEN Analyze 2 THEN Analyze 3
| | |     THEN
| | |     InstHyp [i;t-1] 3
| | |     THEN
| | |     Analyze -1
| | |     THEN
| | |     InstHyp [x@0] -2
| | |     THEN
| | |     NthHypSq -1
| | |     THEN
| | |     RepeatFor 2 Analyze
| | |
| | .\
| | . .....falsecase .....
| | . 1. the_w : World
| | . 2. FairFifo
| | . 3. ∀e,e’:E. loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
| | . 4. ∀e:E. first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
| | . 5. ∀e:E.
| | .    ¬first(e)
| | .    ⇒
| | .    loc(pred(e)) = loc(e) & pred(e) <c e
| | .    & (∀e’:E. loc(e’) = loc(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
| | . 6. t : ℤ
```

```
| | . 7. 0<t
| | . 8. ∀i:Id.
| | .     ¬first(<i,t-1>)
| | .     ⇒
| | .     (∀x:Id. (x when <i,t-1>) = (x after pred(<i,t-1>)))
| | . 9. ¬t = 0
| | . 10. i : Id
| | . 11. ¬isnull(a(i;t-1))
| | . ⊢  ¬false2 ⇒ (∀x:Id. (x when <i,t>) = (x after <i,t-1>))
| | .   by Auto THEN Unfolds [‘w-when‘;‘w-after‘] 0 THEN Reduce 0
| | .      THEN
| | .      Subst’ (t-1+1 = t) 0
| |
| .\
| . 1. the_w : World
| . 2. FairFifo
| . 3. ∀e,e’:E. loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
| . 4. ∀e:E. first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
| . 5. ∀e:E.
| .     ¬first(e)
| .     ⇒
| .     loc(pred(e)) = loc(e) & pred(e) <c e
| .     & (∀e’:E. loc(e’) = loc(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
| . 6. ∀t:ℕ, i:Id.
| .     ¬first(<i,t>)
| .     ⇒
| .     (∀x:Id. (x when <i,t>) = (x after pred(<i,t>)))
| . ⊢  ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| .   by Auto THEN InstHyp [time(e);loc(e);x] 6
| .      THEN
| .      All
| .      (RWO
| .       Thm* ∀the_w:World, e:E. <loc(e),time(e)> ∼ e
| .       {w_DASH_loc_DASH_time})
|
|\
| 1. the_w : World
| 2. FairFifo
| 3. ∀e,e’:E. loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
| 4. ∀e:E. first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
| 5. ∀e:E.
|     ¬first(e)
|     ⇒
|     loc(pred(e)) = loc(e) & pred(e) <c e
|     & (∀e’:E. loc(e’) = loc(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
| 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| ⊢  Trans e,e’:E. e <c e’

|   by Unfold ‘w-causl‘ 0 THEN Reduce 0
|      THEN
|      BackThru
|      Thm* ∀R:(T→T→Prop). Trans x,y:T. x R^+ y    {rel_plus_trans}
|
|\
| 1. the_w : World
| 2. FairFifo
```

```
| 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| 5. ∀e:E.
|    ¬first(e)
|    ⇒
|    loc(pred(e)) = loc(e) & pred(e) <c e
|    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| 7. Trans e,e':E. e <c e'
| ⊢  SWellFounded(e <c e')
|   by Unfold 'strongwellfounded' 0 THEN Reduce 0
|      THEN
|      InstConcl [λe.time(e)]
|      THEN
|      Reduce 0
|      THEN
|      BackThru
|      Thm* ∀the_w:World, e,e':E.
|           FairFifo ⇒ e <c e' ⇒ time(e)<time(e')
|      {w_DASH_causl_DASH_time}
|
|\
| 1. the_w : World
| 2. FairFifo
| 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| 5. ∀e:E.
| |  ¬first(e)
| |  ⇒
| |  loc(pred(e)) = loc(e) & pred(e) <c e
| |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| 7. Trans e,e':E. e <c e'
| 8. SWellFounded(e <c e')
| ⊢  ∀e:E.
| |  isrcv(kind(e))
| |  ⇒
| |  sends(lnk(kind(e));sender(e))[index(e)]
| |  =
| |  msg(lnk(kind(e));tag(kind(e));val(e))
| | by Auto
| |    THEN
| |    AssertBY (∃n:ℕtime(e). match(lnk(kind(e));n;time(e)))
| |    (BackThru
| |     Thm* ∀the_w:World, e:E.
| |          FairFifo
| |          ⇒
| |          isrcv(kind(e))
| |          ⇒
| |          (∃t:ℕtime(e). match(lnk(kind(e));t;time(e)))
| |     {w_DASH_match_DASH_exists})
| |    THEN
| |    Inst
| |    Thm* ∀f:(ℕ→𝔹).
| |         (∃n:ℕ. f(n)) ⇒ f(mu(f)) & (∀i:ℕ. i<mu(f) ⇒ ¬f(i))
```

```
| |    {mu_DASH_property}
| |    [λt.match(lnk(kind(e));t;time(e))]
| |    THENA
| |    (Reduce 0 THEN ParallelOp -1)
| |    THEN
| |    Reduce -1
| |    THEN
| |    ExRepD
| |    THEN
| |    Thin -1
| |    THEN
| |    MoveToConcl -1
| |    THEN
| |    Unfold `w-index` 0
| |    THEN
| |    Unfold `w-sender` 0
| |    THEN
| |    GenConcl (mu(λt.match(lnk(kind(e));t;time(e))) = t)
| |    THENA
| |    (Auto THEN Reduce 0 THEN InstConcl [n])
| |    THEN
| |    Subst (time(<source(lnk(kind(e))),t>) ~ t) 0
| |
| |\
| | .....equality .....
| | 1. the_w : World
| | 2. FairFifo
| | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | 5. ∀e:E.
| |    ¬first(e)
| |    ⇒
| |    loc(pred(e)) = loc(e) & pred(e) <c e
| |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | 7. Trans e,e':E. e <c e'
| | 8. SWellFounded(e <c e')
| | 9. e : E
| | 10. isrcv(kind(e))
| | 11. n : ℕtime(e)
| | 12. match(lnk(kind(e));n;time(e))
| | 13. t : ℕ
| | 14. mu(λt.match(lnk(kind(e));t;time(e))) = t
| | ⊢  time(<source(lnk(kind(e))),t>) ~ t
| |   by Unfold `w-time` 0 THEN Reduce 0
| |
| .\
| . 1. the_w : World
| . 2. FairFifo
| . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . 5. ∀e:E.
| . |  ¬first(e)
| . |  ⇒
| . |  loc(pred(e)) = loc(e) & pred(e) <c e
```

```
| . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . 7. Trans e,e':E. e <c e'
| . 8. SWellFounded(e <c e')
| . 9. e : E
| . 10. isrcv(kind(e))
| . 11. n : ℕtime(e)
| . 12. match(lnk(kind(e));n;time(e))
| . 13. t : ℕ
| . 14. mu(λt.match(lnk(kind(e));t;time(e))) = t
| . ⊢  match(lnk(kind(e));t;time(e))
| . |  ⇒
| . |  sends(lnk(kind(e));<source(lnk(kind(e))),t>)[(||rcvs(...;...)||
| . |  -||snds(lnk(kind(e));t)||)]
| . |  =
| . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | by Analyze 0
| . |    THEN
| . |    Inst Thm: w_DASH_match_DASH_property [the_w;e;t]
| . |    THEN
| . |    ExRepD
| . |
| . 1. the_w : World
| . 2. FairFifo
| . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . 5. ∀e:E.
| . |  ¬first(e)
| . |  ⇒
| . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . 7. Trans e,e':E. e <c e'
| . 8. SWellFounded(e <c e')
| . 9. e : E
| . 10. isrcv(kind(e))
| . 11. n : ℕtime(e)
| . 12. match(lnk(kind(e));n;time(e))
| . 13. t : ℕ
| . 14. mu(λt.match(lnk(kind(e));t;time(e))) = t
| . 15. match(lnk(kind(e));t;time(e))
| . 16. onlnk(lnk(kind(e));m(source(lnk(kind(e)));t))[(||...||
| . |   -||snds(lnk(kind(e));t)||)]
| . |   =
| . |   msg(a(loc(e);time(e)))
| . ⊢  sends(lnk(kind(e));<source(lnk(kind(e))),t>)[(||rcvs(...;...)||
| . |  -||snds(lnk(kind(e));t)||)]
| . |  =
| . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | by Unfold `w-sends` 0 THEN Reduce 0
| . |
| . 1. the_w : World
| . 2. FairFifo
| . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
```

```
| . 5. ∀e:E.
| . |  ¬first(e)
| . |  ⇒
| . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . 7. Trans e,e':E. e <c e'
| . 8. SWellFounded(e <c e')
| . 9. e : E
| . 10. isrcv(kind(e))
| . 11. n : ℕtime(e)
| . 12. match(lnk(kind(e));n;time(e))
| . 13. t : ℕ
| . 14. mu(λt.match(lnk(kind(e));t;time(e))) = t
| . 15. match(lnk(kind(e));t;time(e))
| . 16. onlnk(lnk(kind(e));m(source(lnk(kind(e)));t))[(||...||
| . |   -||snds(lnk(kind(e));t)||)]
| . |   =
| . |   msg(a(loc(e);time(e)))
| . ⊢  onlnk(lnk(kind(e));m(source(lnk(kind(e)));t))[(||...||
| . |  -||snds(lnk(kind(e));t)||)]
| . |  =
| . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | by NthHypSq -1 THEN Analyze THEN Try Trivial
| . |
| . |\
| . | 1. the_w : World
| . | 2. FairFifo
| . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | 5. ∀e:E.
| . |    ¬first(e)
| . |    ⇒
| . |    loc(pred(e)) = loc(e) & pred(e) <c e
| . |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | 7. Trans e,e':E. e <c e'
| . | 8. SWellFounded(e <c e')
| . | 9. e : E
| . | 10. isrcv(kind(e))
| . | 11. n : ℕtime(e)
| . | 12. match(lnk(kind(e));n;time(e))
| . | 13. t : ℕ
| . | 14. mu(λt.match(lnk(kind(e));t;time(e))) = t
| . | 15. match(lnk(kind(e));t;time(e))
| . | 16. onlnk(lnk(kind(e));m(source(lnk(kind(e)));t))[(||...||
| . |     -||snds(lnk(kind(e));t)||)]
| . |     =
| . |     msg(a(loc(e);time(e)))
| . | ⊢  Msg(the_w.M) ~ Msg  by Fold `w-Msg` 0
| . |
| . .\
| . . 1. the_w : World
| . . 2. FairFifo
| . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
```

```
| . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . . 5. ∀e:E.
| . .    ¬first(e)
| . .    ⇒
| . .    loc(pred(e)) = loc(e) & pred(e) <c e
| . .    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . . 7. Trans e,e':E. e <c e'
| . . 8. SWellFounded(e <c e')
| . . 9. e : E
| . . 10. isrcv(kind(e))
| . . 11. n : ℕtime(e)
| . . 12. match(lnk(kind(e));n;time(e))
| . . 13. t : ℕ
| . . 14. mu(λt.match(lnk(kind(e));t;time(e))) = t
| . . 15. match(lnk(kind(e));t;time(e))
| . . 16. onlnk(lnk(kind(e));m(source(lnk(kind(e)));t))[(||...||
| . .     -||snds(lnk(kind(e));t)||)]
| . .     =
| . .     msg(a(loc(e);time(e)))
| . . ⊢  msg(lnk(kind(e));tag(kind(e));val(e)) ~ msg(a(loc(e);...))
| . .   by Unfolds [`w-msg`;`w-ekind`;`w-eval`] 0
| . .      THEN
| . .      Unfolds [`w-act`;`w-loc`;`w-time`] 0
| . .      THEN
| . .      Analyze
|
|\
| 1. the_w : World
| 2. FairFifo
| 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| 5. ∀e:E.
|    ¬first(e)
|    ⇒
|    loc(pred(e)) = loc(e) & pred(e) <c e
|    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| 7. Trans e,e':E. e <c e'
| 8. SWellFounded(e <c e')
| 9. ∀e:E.
|    isrcv(kind(e))
|    ⇒
|    sends(lnk(kind(e));sender(e))[index(e)]
|    =
|    msg(lnk(kind(e));tag(kind(e));val(e))
| ⊢  ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
|   by Auto THEN Unfold `w-causl` 0 THEN Unfold `rel_plus` 0
|      THEN
|      Reduce 0
|      THEN
|      InstConcl [1]
|      THEN
|      RecUnfold `rel_exp` 0
|      THEN
```

```
|       Reduce 0
|       THEN
|       RecUnfold `rel_exp` 0
|       THEN
|       Reduce 0
|       THEN
|       InstConcl [e]
|       THEN
|       OrRight
|
|\
| 1. the_w : World
| 2. FairFifo
| 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| 5. ∀e:E.
| |  ¬first(e)
| |  ⇒
| |  loc(pred(e)) = loc(e) & pred(e) <c e
| |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| 7. Trans e,e':E. e <c e'
| 8. SWellFounded(e <c e')
| 9. ∀e:E.
| |  isrcv(kind(e))
| |  ⇒
| |  sends(lnk(kind(e));sender(e))[index(e)]
| |  =
| |  msg(lnk(kind(e));tag(kind(e));val(e))
| 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| ⊢  ∀e,e':E.
| |  e <c e'
| |  ⇒
| |  ¬first(e') & e <c pred(e') ∨ e = pred(e')
| |  ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | by Auto THEN Unfold `w-causl` -1
| |    THEN
| |    FwdThru
| |    Thm* ∀R:(T→T→Prop), x,y:T.
| |         (x R^+ y) ⇒ (x R y) ∨ (∃z:T. (x R^+ z) & (z R y))
| |    {rel_plus_implies}
| |    [-1]
| |    THEN
| |    Reduce -1
| |    THEN
| |    Fold `w-causl` -1
| |    THEN
| |    Analyze -1
| |
| |\
| | 1. the_w : World
| | 2. FairFifo
| | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | 5. ∀e:E.
```

```
| | |  ¬first(e)
| | |  ⇒
| | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | 7. Trans e,e':E. e <c e'
| | 8. SWellFounded(e <c e')
| | 9. ∀e:E.
| | |  isrcv(kind(e))
| | |  ⇒
| | |  sends(lnk(kind(e));sender(e))[index(e)]
| | |  =
| | |  msg(lnk(kind(e));tag(kind(e));val(e))
| | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | 11. e : E
| | 12. e' : E
| | 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | 14. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')
| | ⊢  ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | |  ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | | by Analyze -1
| | |
| | |\
| | | 1. the_w : World
| | | 2. FairFifo
| | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | 5. ∀e:E.
| | | |  ¬first(e)
| | | |  ⇒
| | | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | | 7. Trans e,e':E. e <c e'
| | | 8. SWellFounded(e <c e')
| | | 9. ∀e:E.
| | | |  isrcv(kind(e))
| | | |  ⇒
| | | |  sends(lnk(kind(e));sender(e))[index(e)]
| | | |  =
| | | |  msg(lnk(kind(e));tag(kind(e));val(e))
| | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | 11. e : E
| | | 12. e' : E
| | | 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | | 14. e <loc e'
| | | ⊢  ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | | |  ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | | | by OrLeft
| | | |
| | | 1. the_w : World
| | | 2. FairFifo
| | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | 5. ∀e:E.
```

```
| | | |  ¬first(e)
| | | |  ⇒
| | | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | | 7. Trans e,e':E. e <c e'
| | | 8. SWellFounded(e <c e')
| | | 9. ∀e:E.
| | | |  isrcv(kind(e))
| | | |  ⇒
| | | |  sends(lnk(kind(e));sender(e))[index(e)]
| | | |  =
| | | |  msg(lnk(kind(e));tag(kind(e));val(e))
| | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | 11. e : E
| | | 12. e' : E
| | | 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | | 14. e <loc e'
| | | ⊢  ¬first(e') & e <c pred(e') ∨ e = pred(e')  by Analyze 0
| | | |
| | | |\
| | | | | 1. the_w : World
| | | | | 2. FairFifo
| | | | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | | | 5. ∀e:E.
| | | | |    ¬first(e)
| | | | |    ⇒
| | | | |    loc(pred(e)) = loc(e) & pred(e) <c e
| | | | |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | | | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | | | | 7. Trans e,e':E. e <c e'
| | | | | 8. SWellFounded(e <c e')
| | | | | 9. ∀e:E.
| | | | |    isrcv(kind(e))
| | | | |    ⇒
| | | | |    sends(lnk(kind(e));sender(e))[index(e)]
| | | | |    =
| | | | |    msg(lnk(kind(e));tag(kind(e));val(e))
| | | | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | | | 11. e : E
| | | | | 12. e' : E
| | | | | 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | | | | 14. e <loc e'
| | | | | ⊢  ¬first(e')
| | | | |   by RWO
| | | | |      Thm* ∀the_w:World, e,e':E.
| | | | |           FairFifo
| | | | |           ⇒
| | | | |           (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| | | | |      {w_DASH_locl_DASH_iff}
| | | | |      -1
| | | | |      THEN
| | | | |      Analyze 0
| | | | |      THEN
```

```
| | | |      InstHyp [e'] 4
| | | |      THEN
| | | |      Analyze -1
| | | |      THEN
| | | |      InstHyp [e] -2
| | | |
| | | .\
| | | . 1. the_w : World
| | | . 2. FairFifo
| | | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | . 5. ∀e:E.
| | | . |  ¬first(e)
| | | . |  ⇒
| | | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | | . 7. Trans e,e':E. e <c e'
| | | . 8. SWellFounded(e <c e')
| | | . 9. ∀e:E.
| | | . |  isrcv(kind(e))
| | | . |  ⇒
| | | . |  sends(lnk(kind(e));sender(e))[index(e)]
| | | . |  =
| | | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | . 11. e : E
| | | . 12. e' : E
| | | . 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | | . 14. e <loc e'
| | | . 15. ¬first(e')
| | | . ⊢  e <c pred(e') ∨ e = pred(e')
| | | . | by Assert (e <c pred(e') ∨ e = pred(e'))
| | | . |
| | | . |\
| | | . | .....assertion .....
| | | . | 1. the_w : World
| | | . | 2. FairFifo
| | | . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | . | 5. ∀e:E.
| | | . | |  ¬first(e)
| | | . | |  ⇒
| | | . | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | | . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | . | 6. ∀e:E.
| | | . | |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | | . | 7. Trans e,e':E. e <c e'
| | | . | 8. SWellFounded(e <c e')
| | | . | 9. ∀e:E.
| | | . | |  isrcv(kind(e))
| | | . | |  ⇒
| | | . | |  sends(lnk(kind(e));sender(e))[index(e)]
| | | . | |  =
| | | . | |  msg(lnk(kind(e));tag(kind(e));val(e))
```

```
| | | . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | . | 11. e : E
| | | . | 12. e' : E
| | | . | 13. e
| | | . | |   λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | | . | 14. e <loc e'
| | | . | 15. ¬first(e')
| | | . | ⊢  e <c pred(e') ∨ e = pred(e')
| | | . | | by InstHyp [e'] 5 THEN Assert (¬pred(e') <loc e)
| | | . | |
| | | . | |\
| | | . | | .....assertion .....
| | | . | | 1. the_w : World
| | | . | | 2. FairFifo
| | | . | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | . | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | . | | 5. ∀e:E.
| | | . | |    ¬first(e)
| | | . | |    ⇒
| | | . | |    loc(pred(e)) = loc(e) & pred(e) <c e
| | | . | |    & (∀e':E.
| | | . | |       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | . | | 6. ∀e:E.
| | | . | |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | | . | | 7. Trans e,e':E. e <c e'
| | | . | | 8. SWellFounded(e <c e')
| | | . | | 9. ∀e:E.
| | | . | |    isrcv(kind(e))
| | | . | |    ⇒
| | | . | |    sends(lnk(kind(e));sender(e))[index(e)]
| | | . | |    =
| | | . | |    msg(lnk(kind(e));tag(kind(e));val(e))
| | | . | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | . | | 11. e : E
| | | . | | 12. e' : E
| | | . | | 13. e
| | | . | |     λe,e'. e <loc e'
| | | . | |     ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | | . | | 14. e <loc e'
| | | . | | 15. ¬first(e')
| | | . | | 16. loc(pred(e')) = loc(e') & pred(e') <c e'
| | | . | |     & (∀e'@0:E.
| | | . | |        loc(e'@0) = loc(e')
| | | . | |        ⇒
| | | . | |        ¬(pred(e') <c e'@0 & e'@0 <c e'))
| | | . | | ⊢  ¬pred(e') <loc e
| | | . | |   by RWO
| | | . | |      Thm* ∀the_w:World, e,e':E.
| | | . | |           FairFifo
| | | . | |           ⇒
| | | . | |           (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| | | . | |      {w_DASH_locl_DASH_iff}
| | | . | |      -3
| | | . | |      THEN
| | | . | |      InstHyp [e] -1
```

```
| | | . | |      THEN
| | | . | |      ParallelOp -1
| | | . | |      THEN
| | | . | |      RWO
| | | . | |      Thm* ∀the_w:World, e,e':E.
| | | . | |           FairFifo
| | | . | |           ⇒
| | | . | |           (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| | | . | |      {w_DASH_locl_DASH_iff}
| | | . | |      -1
| | | . | |
| | | . | .\
| | | . | . 1. the_w : World
| | | . | . 2. FairFifo
| | | . | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | . | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | . | . 5. ∀e:E.
| | | . | . |  ¬first(e)
| | | . | . |  ⇒
| | | . | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | | . | . |  & (∀e':E.
| | | . | . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | . | . 6. ∀e:E.
| | | . | . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | | . | . 7. Trans e,e':E. e <c e'
| | | . | . 8. SWellFounded(e <c e')
| | | . | . 9. ∀e:E.
| | | . | . |  isrcv(kind(e))
| | | . | . |  ⇒
| | | . | . |  sends(lnk(kind(e));sender(e))[index(e)]
| | | . | . |  =
| | | . | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | | . | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | . | . 11. e : E
| | | . | . 12. e' : E
| | | . | . 13. e
| | | . | . |   λe,e'. e <loc e'
| | | . | . |   ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | | . | . 14. e <loc e'
| | | . | . 15. ¬first(e')
| | | . | . 16. loc(pred(e')) = loc(e') & pred(e') <c e'
| | | . | . |   & (∀e'@0:E.
| | | . | . |       loc(e'@0) = loc(e')
| | | . | . |       ⇒
| | | . | . |       ¬(pred(e') <c e'@0 & e'@0 <c e'))
| | | . | . 17. ¬pred(e') <loc e
| | | . | . ⊢  e <c pred(e') ∨ e = pred(e')
| | | . | . | by AllHyps
| | | . | . |    (RWO
| | | . | . |     Thm* ∀the_w:World, e,e':E.
| | | . | . |          FairFifo
| | | . | . |          ⇒
| | | . | . |          (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| | | . | . |     {w_DASH_locl_DASH_iff})
| | | . | . |    THEN
```

```
| | | . | . |    InstHyp [e;pred(e')] 3
| | | . | . |    THEN
| | | . | . |    Analyze -1
| | | . | . |
| | | . | . |\
| | | . | . | 1. the_w : World
| | | . | . | 2. FairFifo
| | | . | . | 3. ∀e,e':E.
| | | . | . |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | . | . | 4. ∀e:E.
| | | . | . |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | . | . | 5. ∀e:E.
| | | . | . |    ¬first(e)
| | | . | . |    ⇒
| | | . | . |    loc(pred(e)) = loc(e) & pred(e) <c e
| | | . | . |    & (∀e':E.
| | | . | . |       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | . | . | 6. ∀e:E.
| | | . | . |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | | . | . | 7. Trans e,e':E. e <c e'
| | | . | . | 8. SWellFounded(e <c e')
| | | . | . | 9. ∀e:E.
| | | . | . |    isrcv(kind(e))
| | | . | . |    ⇒
| | | . | . |    sends(lnk(kind(e));sender(e))[index(e)]
| | | . | . |    =
| | | . | . |    msg(lnk(kind(e));tag(kind(e));val(e))
| | | . | . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | . | . | 11. e : E
| | | . | . | 12. e' : E
| | | . | . | 13. e
| | | . | . |     λe,e'. e <loc e'
| | | . | . |     ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | | . | . | 14. loc(e) = loc(e')
| | | . | . | 15. e <c e'
| | | . | . | 16. ¬first(e')
| | | . | . | 17. loc(pred(e')) = loc(e')
| | | . | . | 18. pred(e') <c e'
| | | . | . | 19. ∀e'@0:E.
| | | . | . |     loc(e'@0) = loc(e')
| | | . | . |     ⇒
| | | . | . |     ¬(pred(e') <c e'@0 & e'@0 <c e')
| | | . | . | 20. ¬(loc(pred(e')) = loc(e) & pred(e') <c e)
| | | . | . | 21. e <c pred(e')
| | | . | . | ⊢  e <c pred(e') ∨ e = pred(e')  by OrLeft
| | | . | . |
| | | . | . .\
| | | . | . . 1. the_w : World
| | | . | . . 2. FairFifo
| | | . | . . 3. ∀e,e':E.
| | | . | . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | . | . . 4. ∀e:E.
| | | . | . . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | . | . . 5. ∀e:E.
| | | . | . . |  ¬first(e)
```

```
| | | . | . . |  ⇒
| | | . | . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | | . | . . |  & (∀e':E.
| | | . | . . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | . | . . 6. ∀e:E.
| | | . | . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | | . | . . 7. Trans e,e':E. e <c e'
| | | . | . . 8. SWellFounded(e <c e')
| | | . | . . 9. ∀e:E.
| | | . | . . |  isrcv(kind(e))
| | | . | . . |  ⇒
| | | . | . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | | . | . . |  =
| | | . | . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | | . | . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | . | . . 11. e : E
| | | . | . . 12. e' : E
| | | . | . . 13. e
| | | . | . . |   λe,e'. e <loc e'
| | | . | . . |   ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | | . | . . 14. loc(e) = loc(e')
| | | . | . . 15. e <c e'
| | | . | . . 16. ¬first(e')
| | | . | . . 17. loc(pred(e')) = loc(e')
| | | . | . . 18. pred(e') <c e'
| | | . | . . 19. ∀e'@0:E.
| | | . | . . |   loc(e'@0) = loc(e')
| | | . | . . |   ⇒
| | | . | . . |   ¬(pred(e') <c e'@0 & e'@0 <c e')
| | | . | . . 20. ¬(loc(pred(e')) = loc(e) & pred(e') <c e)
| | | . | . . 21. e = pred(e') ∨ pred(e') <c e
| | | . | . . ⊢  e <c pred(e') ∨ e = pred(e')  by Analyze -1

| | | . | . . |
| | | . | . . |\
| | | . | . . | 1. the_w : World
| | | . | . . | 2. FairFifo
| | | . | . . | 3. ∀e,e':E.
| | | . | . . |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | . | . . | 4. ∀e:E.
| | | . | . . |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | . | . . | 5. ∀e:E.
| | | . | . . |    ¬first(e)
| | | . | . . |    ⇒
| | | . | . . |    loc(pred(e)) = loc(e) & pred(e) <c e
| | | . | . . |    & (∀e':E.
| | | . | . . |       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | . | . . | 6. ∀e:E.
| | | . | . . |    ¬first(e)
| | | . | . . |    ⇒
| | | . | . . |    (∀x:Id. (x when e) = (x after pred(e)))
| | | . | . . | 7. Trans e,e':E. e <c e'
| | | . | . . | 8. SWellFounded(e <c e')
| | | . | . . | 9. ∀e:E.
| | | . | . . |    isrcv(kind(e))
| | | . | . . |    ⇒
```

```
| | | . | . . |    sends(lnk(kind(e));sender(e))[index(e)]
| | | . | . . |    =
| | | . | . . |    msg(lnk(kind(e));tag(kind(e));val(e))
| | | . | . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | . | . . | 11. e : E
| | | . | . . | 12. e' : E
| | | . | . . | 13. e
| | | . | . . |     λe,e'. e <loc e'
| | | . | . . |     ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | | . | . . | 14. loc(e) = loc(e')
| | | . | . . | 15. e <c e'
| | | . | . . | 16. ¬first(e')
| | | . | . . | 17. loc(pred(e')) = loc(e')
| | | . | . . | 18. pred(e') <c e'
| | | . | . . | 19. ∀e'@0:E.
| | | . | . . |     loc(e'@0) = loc(e')
| | | . | . . |     ⇒
| | | . | . . |     ¬(pred(e') <c e'@0 & e'@0 <c e')
| | | . | . . | 20. ¬(loc(pred(e')) = loc(e) & pred(e') <c e)
| | | . | . . | 21. e = pred(e')
| | | . | . . | ⊢  e <c pred(e') ∨ e = pred(e')  by OrRight
| | | . | . . |
| | | . | . . .\
| | | . | . . . 1. the_w : World
| | | . | . . . 2. FairFifo
| | | . | . . . 3. ∀e,e':E.
| | | . | . . .    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | . | . . . 4. ∀e:E.
| | | . | . . .    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | . | . . . 5. ∀e:E.
| | | . | . . .    ¬first(e)
| | | . | . . .    ⇒
| | | . | . . .    loc(pred(e)) = loc(e) & pred(e) <c e
| | | . | . . .    & (∀e':E.
| | | . | . . .       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | . | . . . 6. ∀e:E.
| | | . | . . .    ¬first(e)
| | | . | . . .    ⇒
| | | . | . . .    (∀x:Id. (x when e) = (x after pred(e)))
| | | . | . . . 7. Trans e,e':E. e <c e'
| | | . | . . . 8. SWellFounded(e <c e')
| | | . | . . . 9. ∀e:E.
| | | . | . . .    isrcv(kind(e))
| | | . | . . .    ⇒
| | | . | . . .    sends(lnk(kind(e));sender(e))[index(e)]
| | | . | . . .    =
| | | . | . . .    msg(lnk(kind(e));tag(kind(e));val(e))
| | | . | . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | . | . . . 11. e : E
| | | . | . . . 12. e' : E
| | | . | . . . 13. e
| | | . | . . .     λe,e'. e <loc e'
| | | . | . . .     ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | | . | . . . 14. loc(e) = loc(e')
| | | . | . . . 15. e <c e'
```

```
| | | . | . . . 16. ¬first(e’)
| | | . | . . . 17. loc(pred(e’)) = loc(e’)
| | | . | . . . 18. pred(e’) <c e’
| | | . | . . . 19. ∀e’@0:E.
| | | . | . . .     loc(e’@0) = loc(e’)
| | | . | . . .     ⇒
| | | . | . . .     ¬(pred(e’) <c e’@0 & e’@0 <c e’)
| | | . | . . . 20. ¬(loc(pred(e’)) = loc(e) & pred(e’) <c e)
| | | . | . . . 21. pred(e’) <c e
| | | . | . . . ⊢  e <c pred(e’) ∨ e = pred(e’)  by Analyze -2
| | | . |
| | | . .\
| | | . . 1. the_w : World
| | | . . 2. FairFifo
| | | . . 3. ∀e,e’:E. loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
| | | . . 4. ∀e:E. first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
| | | . . 5. ∀e:E.
| | | . .    ¬first(e)
| | | . .    ⇒
| | | . .    loc(pred(e)) = loc(e) & pred(e) <c e
| | | . .    & (∀e’:E. loc(e’) = loc(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
| | | . . 6. ∀e:E.
| | | . .    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | | . . 7. Trans e,e’:E. e <c e’
| | | . . 8. SWellFounded(e <c e’)
| | | . . 9. ∀e:E.
| | | . .    isrcv(kind(e))
| | | . .    ⇒
| | | . .    sends(lnk(kind(e));sender(e))[index(e)]
| | | . .    =
| | | . .    msg(lnk(kind(e));tag(kind(e));val(e))
| | | . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | . . 11. e : E
| | | . . 12. e’ : E
| | | . . 13. e
| | | . .     λe,e’. e <loc e’ ∨ isrcv(kind(e’)) & e = sender(e’)^+ e’
| | | . . 14. e <loc e’
| | | . . 15. ¬first(e’)
| | | . . 16. e <c pred(e’) ∨ e = pred(e’)
| | | . . ⊢  e <c pred(e’) ∨ e = pred(e’)  by ParallelOp -1
| | |
| | .\
| | . 1. the_w : World
| | . 2. FairFifo
| | . 3. ∀e,e’:E. loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
| | . 4. ∀e:E. first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
| | . 5. ∀e:E.
| | .    ¬first(e)
| | .    ⇒
| | .    loc(pred(e)) = loc(e) & pred(e) <c e
| | .    & (∀e’:E. loc(e’) = loc(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
| | . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . 7. Trans e,e’:E. e <c e’
| | . 8. SWellFounded(e <c e’)
| | . 9. ∀e:E.
```

```
| | .    isrcv(kind(e))
| | .    ⇒
| | .    sends(lnk(kind(e));sender(e))[index(e)]
| | .    =
| | .    msg(lnk(kind(e));tag(kind(e));val(e))
| | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . 11. e : E
| | . 12. e' : E
| | . 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| | . 14. isrcv(kind(e')) & e = sender(e')
| | . ⊢  ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | .    ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | .   by ExRepD THEN OrRight THEN OrRight
| |
| .\
| . 1. the_w : World
| . 2. FairFifo
| . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . 5. ∀e:E.
| . |  ¬first(e)
| . |  ⇒
| . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . 7. Trans e,e':E. e <c e'
| . 8. SWellFounded(e <c e')
| . 9. ∀e:E.
| . |  isrcv(kind(e))
| . |  ⇒
| . |  sends(lnk(kind(e));sender(e))[index(e)]
| . |  =
| . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . 11. e : E
| . 12. e' : E
| . 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . 14. ∃z:E. e <c z & z <loc e' ∨ isrcv(kind(e')) & z = sender(e')
| . ⊢  ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . |  ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | by ExRepD THEN Analyze -1 THEN ExRepD
| . |
| . |\
| . | 1. the_w : World
| . | 2. FairFifo
| . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | 5. ∀e:E.
| . | |  ¬first(e)
| . | |  ⇒
| . | |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | 7. Trans e,e':E. e <c e'
| . | 8. SWellFounded(e <c e')
```

```
| . | 9. ∀e:E.
| . | |  isrcv(kind(e))
| . | |  ⇒
| . | |  sends(lnk(kind(e));sender(e))[index(e)]
| . | |  =
| . | |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | 11. e : E
| . | 12. e' : E
| . | 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | 14. z : E
| . | 15. e <c z
| . | 16. z <loc e'
| . | ⊢  ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | |  ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | | by OrLeft
| . | |
| . | 1. the_w : World
| . | 2. FairFifo
| . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | 5. ∀e:E.
| . | |  ¬first(e)
| . | |  ⇒
| . | |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | 7. Trans e,e':E. e <c e'
| . | 8. SWellFounded(e <c e')
| . | 9. ∀e:E.
| . | |  isrcv(kind(e))
| . | |  ⇒
| . | |  sends(lnk(kind(e));sender(e))[index(e)]
| . | |  =
| . | |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | 11. e : E
| . | 12. e' : E
| . | 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | 14. z : E
| . | 15. e <c z
| . | 16. z <loc e'
| . | ⊢  ¬first(e') & e <c pred(e') ∨ e = pred(e')  by Analyze 0
| . | |
| . | |\
| . | | 1. the_w : World
| . | | 2. FairFifo
| . | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | | 5. ∀e:E.
| . | |    ¬first(e)
| . | |    ⇒
| . | |    loc(pred(e)) = loc(e) & pred(e) <c e
| . | |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
```

```
| . | | 7. Trans e,e':E. e <c e'
| . | | 8. SWellFounded(e <c e')
| . | | 9. ∀e:E.
| . | |    isrcv(kind(e))
| . | |    ⇒
| . | |    sends(lnk(kind(e));sender(e))[index(e)]
| . | |    =
| . | |    msg(lnk(kind(e));tag(kind(e));val(e))
| . | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | | 11. e : E
| . | | 12. e' : E
| . | | 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | | 14. z : E
| . | | 15. e <c z
| . | | 16. z <loc e'
| . | | ⊢  ¬first(e')
| . | |   by RWO
| . | |      Thm* ∀the_w:World, e,e':E.
| . | |           FairFifo
| . | |           ⇒
| . | |           (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| . | |      {w_DASH_locl_DASH_iff}
| . | |      -1
| . | |      THEN
| . | |      Analyze 0
| . | |      THEN
| . | |      InstHyp [e'] 4
| . | |      THEN
| . | |      Analyze -1
| . | |      THEN
| . | |      InstHyp [z] -2
| . | |
| . | .\
| . | . 1. the_w : World
| . | . 2. FairFifo
| . | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . 5. ∀e:E.
| . | . |  ¬first(e)
| . | . |  ⇒
| . | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | . 7. Trans e,e':E. e <c e'
| . | . 8. SWellFounded(e <c e')
| . | . 9. ∀e:E.
| . | . |  isrcv(kind(e))
| . | . |  ⇒
| . | . |  sends(lnk(kind(e));sender(e))[index(e)]
| . | . |  =
| . | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . 11. e : E
| . | . 12. e' : E
| . | . 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
```

```
| . | . 14. z : E
| . | . 15. e <c z
| . | . 16. z <loc e'
| . | . 17. ¬first(e')
| . | . ⊢  e <c pred(e') ∨ e = pred(e')  by OrLeft
| . | . |
| . | . 1. the_w : World
| . | . 2. FairFifo
| . | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . 5. ∀e:E.
| . | . |  ¬first(e)
| . | . |  ⇒
| . | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | . 7. Trans e,e':E. e <c e'
| . | . 8. SWellFounded(e <c e')
| . | . 9. ∀e:E.
| . | . |  isrcv(kind(e))
| . | . |  ⇒
| . | . |  sends(lnk(kind(e));sender(e))[index(e)]
| . | . |  =
| . | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . 11. e : E
| . | . 12. e' : E
| . | . 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | . 14. z : E
| . | . 15. e <c z
| . | . 16. z <loc e'
| . | . 17. ¬first(e')
| . | . ⊢  e <c pred(e')  by Assert (z <c pred(e') ∨ z = pred(e'))
| . | . |
| . | . |\
| . | . | .....assertion .....
| . | . | 1. the_w : World
| . | . | 2. FairFifo
| . | . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . | 5. ∀e:E.
| . | . | |  ¬first(e)
| . | . | |  ⇒
| . | . | |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . | 6. ∀e:E.
| . | . | |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | . | 7. Trans e,e':E. e <c e'
| . | . | 8. SWellFounded(e <c e')
| . | . | 9. ∀e:E.
| . | . | |  isrcv(kind(e))
| . | . | |  ⇒
| . | . | |  sends(lnk(kind(e));sender(e))[index(e)]
| . | . | |  =
| . | . | |  msg(lnk(kind(e));tag(kind(e));val(e))
```

```
| . | . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . | 11. e : E
| . | . | 12. e' : E
| . | . | 13. e
| . | . | |   λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | . | 14. z : E
| . | . | 15. e <c z
| . | . | 16. z <loc e'
| . | . | 17. ¬first(e')
| . | . | ⊢  z <c pred(e') ∨ z = pred(e')
| . | . | | by InstHyp [e'] 5 THEN Assert (¬pred(e') <loc z)
| . | . | |
| . | . | |\
| . | . | | .....assertion .....
| . | . | | 1. the_w : World
| . | . | | 2. FairFifo
| . | . | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . | | 5. ∀e:E.
| . | . | |    ¬first(e)
| . | . | |    ⇒
| . | . | |    loc(pred(e)) = loc(e) & pred(e) <c e
| . | . | |    & (∀e':E.
| . | . | |       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . | | 6. ∀e:E.
| . | . | |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | . | | 7. Trans e,e':E. e <c e'
| . | . | | 8. SWellFounded(e <c e')
| . | . | | 9. ∀e:E.
| . | . | |    isrcv(kind(e))
| . | . | |    ⇒
| . | . | |    sends(lnk(kind(e));sender(e))[index(e)]
| . | . | |    =
| . | . | |    msg(lnk(kind(e));tag(kind(e));val(e))
| . | . | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . | | 11. e : E
| . | . | | 12. e' : E
| . | . | | 13. e
| . | . | |     λe,e'. e <loc e'
| . | . | |     ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | . | | 14. z : E
| . | . | | 15. e <c z
| . | . | | 16. z <loc e'
| . | . | | 17. ¬first(e')
| . | . | | 18. loc(pred(e')) = loc(e') & pred(e') <c e'
| . | . | |     & (∀e'@0:E.
| . | . | |        loc(e'@0) = loc(e')

| . | . | |        ⇒
| . | . | |        ¬(pred(e') <c e'@0 & e'@0 <c e'))
| . | . | | ⊢  ¬pred(e') <loc z
| . | . | |   by RWO
| . | . | |      Thm* ∀the_w:World, e,e':E.
| . | . | |           FairFifo
| . | . | |           ⇒
| . | . | |           (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
```

```
| . | . | |      {w_DASH_locl_DASH_iff}
| . | . | |      -3
| . | . | |      THEN
| . | . | |      InstHyp [z] -1
| . | . | |      THEN
| . | . | |      ParallelOp -1
| . | . | |      THEN
| . | . | |      RWO
| . | . | |      Thm* ∀the_w:World, e,e':E.
| . | . | |           FairFifo
| . | . | |           ⇒
| . | . | |           (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| . | . | |      {w_DASH_locl_DASH_iff}
| . | . | |      -1
| . | . | |
| . | . | .\
| . | . | . 1. the_w : World
| . | . | . 2. FairFifo
| . | . | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . | . 5. ∀e:E.
| . | . | . |  ¬first(e)
| . | . | . |  ⇒
| . | . | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | . | . |  & (∀e':E.
| . | . | . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . | . 6. ∀e:E.
| . | . | . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | . | . 7. Trans e,e':E. e <c e'
| . | . | . 8. SWellFounded(e <c e')
| . | . | . 9. ∀e:E.
| . | . | . |  isrcv(kind(e))
| . | . | . |  ⇒
| . | . | . |  sends(lnk(kind(e));sender(e))[index(e)]
| . | . | . |  =
| . | . | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | . | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . | . 11. e : E
| . | . | . 12. e' : E
| . | . | . 13. e
| . | . | . |   λe,e'. e <loc e'
| . | . | . |   ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | . | . 14. z : E
| . | . | . 15. e <c z
| . | . | . 16. z <loc e'
| . | . | . 17. ¬first(e')
| . | . | . 18. loc(pred(e')) = loc(e') & pred(e') <c e'
| . | . | . |   & (∀e'@0:E.
| . | . | . |       loc(e'@0) = loc(e')
| . | . | . |       ⇒
| . | . | . |       ¬(pred(e') <c e'@0 & e'@0 <c e'))
| . | . | . 19. ¬pred(e') <loc z
| . | . | . ⊢  z <c pred(e') ∨ z = pred(e')
| . | . | . | by AllHyps
| . | . | . |    (RWO
```

```
| . | . | . |     Thm* ∀the_w:World, e,e':E.
| . | . | . |          FairFifo
| . | . | . |          ⇒
| . | . | . |          (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| . | . | . |     {w_DASH_locl_DASH_iff})
| . | . | . |    THEN
| . | . | . |    InstHyp [z;pred(e')] 3
| . | . | . |    THEN
| . | . | . |    Analyze -1
| . | . | . |
| . | . | . |\
| . | . | . | 1. the_w : World
| . | . | . | 2. FairFifo
| . | . | . | 3. ∀e,e':E.
| . | . | . |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . | . | 4. ∀e:E.
| . | . | . |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . | . | 5. ∀e:E.
| . | . | . |    ¬first(e)
| . | . | . |    ⇒
| . | . | . |    loc(pred(e)) = loc(e) & pred(e) <c e
| . | . | . |    & (∀e':E.
| . | . | . |       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . | . | 6. ∀e:E.
| . | . | . |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | . | . | 7. Trans e,e':E. e <c e'
| . | . | . | 8. SWellFounded(e <c e')
| . | . | . | 9. ∀e:E.
| . | . | . |    isrcv(kind(e))
| . | . | . |    ⇒
| . | . | . |    sends(lnk(kind(e));sender(e))[index(e)]
| . | . | . |    =
| . | . | . |    msg(lnk(kind(e));tag(kind(e));val(e))
| . | . | . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . | . | 11. e : E
| . | . | . | 12. e' : E
| . | . | . | 13. e
| . | . | . |     λe,e'. e <loc e'
| . | . | . |     ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | . | . | 14. z : E
| . | . | . | 15. e <c z
| . | . | . | 16. loc(z) = loc(e')
| . | . | . | 17. z <c e'
| . | . | . | 18. ¬first(e')
| . | . | . | 19. loc(pred(e')) = loc(e')
| . | . | . | 20. pred(e') <c e'
| . | . | . | 21. ∀e'@0:E.
| . | . | . |     loc(e'@0) = loc(e')
| . | . | . |     ⇒
| . | . | . |     ¬(pred(e') <c e'@0 & e'@0 <c e')
| . | . | . | 22. ¬(loc(pred(e')) = loc(z) & pred(e') <c z)
| . | . | . | 23. z <c pred(e')
| . | . | . | ⊢  z <c pred(e') ∨ z = pred(e')  by OrLeft
| . | . | . |
| . | . | . .\
```

```
| . | . | . . 1. the_w : World
| . | . | . . 2. FairFifo
| . | . | . . 3. ∀e,e':E.
| . | . | . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . | . . 4. ∀e:E.
| . | . | . . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . | . . 5. ∀e:E.
| . | . | . . |  ¬first(e)
| . | . | . . |  ⇒
| . | . | . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | . | . . |  & (∀e':E.
| . | . | . . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . | . . 6. ∀e:E.
| . | . | . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | . | . . 7. Trans e,e':E. e <c e'
| . | . | . . 8. SWellFounded(e <c e')
| . | . | . . 9. ∀e:E.
| . | . | . . |  isrcv(kind(e))
| . | . | . . |  ⇒
| . | . | . . |  sends(lnk(kind(e));sender(e))[index(e)]
| . | . | . . |  =
| . | . | . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | . | . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . | . . 11. e : E
| . | . | . . 12. e' : E
| . | . | . . 13. e
| . | . | . . |   λe,e'. e <loc e'
| . | . | . . |   ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | . | . . 14. z : E
| . | . | . . 15. e <c z
| . | . | . . 16. loc(z) = loc(e')
| . | . | . . 17. z <c e'
| . | . | . . 18. ¬first(e')
| . | . | . . 19. loc(pred(e')) = loc(e')
| . | . | . . 20. pred(e') <c e'
| . | . | . . 21. ∀e'@0:E.
| . | . | . . |   loc(e'@0) = loc(e')
| . | . | . . |   ⇒
| . | . | . . |   ¬(pred(e') <c e'@0 & e'@0 <c e')
| . | . | . . 22. ¬(loc(pred(e')) = loc(z) & pred(e') <c z)
| . | . | . . 23. z = pred(e') ∨ pred(e') <c z
| . | . | . . ⊢  z <c pred(e') ∨ z = pred(e')  by Analyze -1
| . | . | . . |
| . | . | . . |\
| . | . | . . | 1. the_w : World
| . | . | . . | 2. FairFifo
| . | . | . . | 3. ∀e,e':E.
| . | . | . . |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . | . . | 4. ∀e:E.
| . | . | . . |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . | . . | 5. ∀e:E.
| . | . | . . |    ¬first(e)
| . | . | . . |    ⇒
| . | . | . . |    loc(pred(e)) = loc(e) & pred(e) <c e
| . | . | . . |    & (∀e':E.
```

```
| . | . | . . . |       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . | . . . | 6. ∀e:E.
| . | . | . . . |    ¬first(e)
| . | . | . . . |    ⇒
| . | . | . . . |    (∀x:Id. (x when e) = (x after pred(e)))
| . | . | . . . | 7. Trans e,e':E. e <c e'
| . | . | . . . | 8. SWellFounded(e <c e')
| . | . | . . . | 9. ∀e:E.
| . | . | . . . |    isrcv(kind(e))
| . | . | . . . |    ⇒
| . | . | . . . |    sends(lnk(kind(e));sender(e))[index(e)]
| . | . | . . . |    =
| . | . | . . . |    msg(lnk(kind(e));tag(kind(e));val(e))
| . | . | . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . | . . . | 11. e : E
| . | . | . . . | 12. e' : E
| . | . | . . . | 13. e
| . | . | . . . |     λe,e'. e <loc e'
| . | . | . . . |     ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | . | . . . | 14. z : E
| . | . | . . . | 15. e <c z
| . | . | . . . | 16. loc(z) = loc(e')
| . | . | . . . | 17. z <c e'
| . | . | . . . | 18. ¬first(e')
| . | . | . . . | 19. loc(pred(e')) = loc(e')
| . | . | . . . | 20. pred(e') <c e'
| . | . | . . . | 21. ∀e'@0:E.
| . | . | . . . |     loc(e'@0) = loc(e')
| . | . | . . . |     ⇒
| . | . | . . . |     ¬(pred(e') <c e'@0 & e'@0 <c e')
| . | . | . . . | 22. ¬(loc(pred(e')) = loc(z) & pred(e') <c z)
| . | . | . . . | 23. z = pred(e')
| . | . | . . . | ⊢  z <c pred(e') ∨ z = pred(e')  by OrRight
| . | . | . . . |
| . | . | . . . .\
| . | . | . . . . 1. the_w : World
| . | . | . . . . 2. FairFifo
| . | . | . . . . 3. ∀e,e':E.
| . | . | . . . .    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . | . . . . 4. ∀e:E.
| . | . | . . . .    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . | . . . . 5. ∀e:E.
| . | . | . . . .    ¬first(e)
| . | . | . . . .    ⇒
| . | . | . . . .    loc(pred(e)) = loc(e) & pred(e) <c e
| . | . | . . . .    & (∀e':E.
| . | . | . . . .       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . | . . . . 6. ∀e:E.
| . | . | . . . .    ¬first(e)
| . | . | . . . .    ⇒
| . | . | . . . .    (∀x:Id. (x when e) = (x after pred(e)))
| . | . | . . . . 7. Trans e,e':E. e <c e'
| . | . | . . . . 8. SWellFounded(e <c e')
| . | . | . . . . 9. ∀e:E.
| . | . | . . . .    isrcv(kind(e))
```

```
| . | . | . . .    ⇒
| . | . | . . .    sends(lnk(kind(e));sender(e))[index(e)]
| . | . | . . .    =
| . | . | . . .    msg(lnk(kind(e));tag(kind(e));val(e))
| . | . | . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . | . . . 11. e : E
| . | . | . . . 12. e' : E
| . | . | . . . 13. e
| . | . | . . .     λe,e'. e <loc e'
| . | . | . . .     ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | . | . . . 14. z : E
| . | . | . . . 15. e <c z
| . | . | . . . 16. loc(z) = loc(e')
| . | . | . . . 17. z <c e'
| . | . | . . . 18. ¬first(e')
| . | . | . . . 19. loc(pred(e')) = loc(e')
| . | . | . . . 20. pred(e') <c e'
| . | . | . . . 21. ∀e'@0:E.
| . | . | . . .     loc(e'@0) = loc(e')
| . | . | . . .     ⇒
| . | . | . . .     ¬(pred(e') <c e'@0 & e'@0 <c e')
| . | . | . . . 22. ¬(loc(pred(e')) = loc(z) & pred(e') <c z)
| . | . | . . . 23. pred(e') <c z
| . | . | . . . ⊢  z <c pred(e') ∨ z = pred(e')  by Analyze -2
| . | . |
| . | . .\
| . | . . 1. the_w : World
| . | . . 2. FairFifo
| . | . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . . 5. ∀e:E.
| . | . . |  ¬first(e)
| . | . . |  ⇒
| . | . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | . . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . . 6. ∀e:E.
| . | . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | . . 7. Trans e,e':E. e <c e'
| . | . . 8. SWellFounded(e <c e')
| . | . . 9. ∀e:E.
| . | . . |  isrcv(kind(e))
| . | . . |  ⇒
| . | . . |  sends(lnk(kind(e));sender(e))[index(e)]
| . | . . |  =
| . | . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . . 11. e : E
| . | . . 12. e' : E
| . | . . 13. e
| . | . . |   λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | . . 14. z : E
| . | . . 15. e <c z
| . | . . 16. z <loc e'
| . | . . 17. ¬first(e')
| . | . . 18. z <c pred(e') ∨ z = pred(e')
```

```
| . | . . ⊢  e <c pred(e')  by Analyze -1
| . | . . |
| . | . . |\
| . | . . | 1. the_w : World
| . | . . | 2. FairFifo
| . | . . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . . | 5. ∀e:E.
| . | . . |    ¬first(e)
| . | . . |    ⇒
| . | . . |    loc(pred(e)) = loc(e) & pred(e) <c e
| . | . . |    & (∀e':E.
| . | . . |       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . . | 6. ∀e:E.
| . | . . |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | . . | 7. Trans e,e':E. e <c e'
| . | . . | 8. SWellFounded(e <c e')
| . | . . | 9. ∀e:E.
| . | . . |    isrcv(kind(e))
| . | . . |    ⇒
| . | . . |    sends(lnk(kind(e));sender(e))[index(e)]
| . | . . |    =
| . | . . |    msg(lnk(kind(e));tag(kind(e));val(e))
| . | . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . . | 11. e : E
| . | . . | 12. e' : E
| . | . . | 13. e
| . | . . |     λe,e'. e <loc e'
| . | . . |     ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | . . | 14. z : E
| . | . . | 15. e <c z
| . | . . | 16. z <loc e'
| . | . . | 17. ¬first(e')
| . | . . | 18. z <c pred(e')
| . | . . | ⊢  e <c pred(e')  by UseTrans z
| . | . . |
| . | . . .\
| . | . . . 1. the_w : World
| . | . . . 2. FairFifo
| . | . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . . . 5. ∀e:E.
| . | . . .    ¬first(e)
| . | . . .    ⇒
| . | . . .    loc(pred(e)) = loc(e) & pred(e) <c e
| . | . . .    & (∀e':E.
| . | . . .       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . . . 6. ∀e:E.
| . | . . .    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | . . . 7. Trans e,e':E. e <c e'
| . | . . . 8. SWellFounded(e <c e')
| . | . . . 9. ∀e:E.
| . | . . .    isrcv(kind(e))
| . | . . .    ⇒
| . | . . .    sends(lnk(kind(e));sender(e))[index(e)]
```

```
| . | . . .    =
| . | . . .    msg(lnk(kind(e));tag(kind(e));val(e))
| . | . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . . . 11. e : E
| . | . . . 12. e' : E
| . | . . . 13. e
| . | . . .     λe,e'. e <loc e'
| . | . . .     ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . | . . . 14. z : E
| . | . . . 15. e <c z
| . | . . . 16. z <loc e'
| . | . . . 17. ¬first(e')
| . | . . . 18. z = pred(e')
| . | . . . ⊢  e <c pred(e')  by RevHypSubst -1 0
| . |
| . .\
| . . 1. the_w : World
| . . 2. FairFifo
| . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . . 5. ∀e:E.
| . .    ¬first(e)
| . .    ⇒
| . .    loc(pred(e)) = loc(e) & pred(e) <c e
| . .    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . . 7. Trans e,e':E. e <c e'
| . . 8. SWellFounded(e <c e')
| . . 9. ∀e:E.
| . .    isrcv(kind(e))
| . .    ⇒
| . .    sends(lnk(kind(e));sender(e))[index(e)]
| . .    =
| . .    msg(lnk(kind(e));tag(kind(e));val(e))
| . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . . 11. e : E
| . . 12. e' : E
| . . 13. e λe,e'. e <loc e' ∨ isrcv(kind(e')) & e = sender(e')^+ e'
| . . 14. z : E
| . . 15. e <c z
| . . 16. isrcv(kind(e'))
| . . 17. z = sender(e')
| . . ⊢  ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . .    ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . .   by OrRight THEN OrLeft THEN HypSubst -1 -3
|
|\
| 1. the_w : World
| 2. FairFifo
| 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| 5. ∀e:E.
| |  ¬first(e)
| |  ⇒
| |  loc(pred(e)) = loc(e) & pred(e) <c e
```

```
| |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| 7. Trans e,e':E. e <c e'
| 8. SWellFounded(e <c e')
| 9. ∀e:E.
| |  isrcv(kind(e))
| |  ⇒
| |  sends(lnk(kind(e));sender(e))[index(e)]
| |  =
| |  msg(lnk(kind(e));tag(kind(e));val(e))
| 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| 11. ∀e,e':E.
| |   e <c e'
| |   ⇒
| |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| ⊢  ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | by Unfold `fair-fifo` 2 THEN ExRepD THEN Analyze -2
| |    THEN
| |    Analyze -3
| |    THEN
| |    All (Unfolds [`w-loc`;`w-time`;`w-ekind`])
| |    THEN
| |    All Reduce
| |    THEN
| |    All (Unfold `w-act`)
| |    THEN
| |    All Reduce
| |
| 1. the_w : World
| 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| 3. ∀i:Id, t:ℕ.
| |  isnull(a(i;t)) ⇒ (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| 4. ∀i:Id, t:ℕ, l:IdLnk.
| |  isrcv(l;a(i;t))
| |  ⇒
| |  destination(l) = i
| |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| 5. ∀l:IdLnk, t:ℕ.
| |  ∃t':ℕ. t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| 8. ∀e:E.
| |  ¬first(e)
| |  ⇒
| |  1of(pred(e)) = 1of(e) & pred(e) <c e
| |  & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| 10. Trans e,e':E. e <c e'
| 11. SWellFounded(e <c e')
| 12. ∀e:E.
| |   isrcv(kind(a(1of(e);2of(e))))
| |   ⇒
| |   sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| |   =
```

```
| |   msg(lnk(kind(a(1of(e);2of(e))));tag(kind(a(1of(e);2of(
| |                                                   e))));val(e))
| 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| 14. ∀e,e':E.
| |   e <c e'
| |   ⇒
| |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| |   ∨ isrcv(kind(a(1of(e');2of(e'))))
| |     & e <c sender(e') ∨ e = sender(e')
| 15. e1 : Id
| 16. e2 : ℕ
| 17. ¬isnull(a(e1;e2))
| 18. isrcv(kind(a(e1;e2)))
| ⊢  e1 = destination(lnk(kind(a(e1;e2))))
| | by InstHyp [e1;e2;lnk(kind(a(e1;e2)))] 4
| |
| .....antecedent .....
| 1. the_w : World
| 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| 3. ∀i:Id, t:ℕ.
| |  isnull(a(i;t)) ⇒ (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| 4. ∀i:Id, t:ℕ, l:IdLnk.
| |  isrcv(l;a(i;t))
| |  ⇒
| |  destination(l) = i
| |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| 5. ∀l:IdLnk, t:ℕ.
| |  ∃t':ℕ. t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| 8. ∀e:E.
| |  ¬first(e)
| |  ⇒
| |  1of(pred(e)) = 1of(e) & pred(e) <c e
| |  & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| 10. Trans e,e':E. e <c e'
| 11. SWellFounded(e <c e')
| 12. ∀e:E.
| |   isrcv(kind(a(1of(e);2of(e))))
| |   ⇒
| |   sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| |   =
```

```
| |   msg(lnk(kind(a(1of(e);2of(e))));tag(kind(a(1of(e);2of(
| |                                                   e))));val(e))
| 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| 14. ∀e,e':E.
| |   e <c e'
| |   ⇒
| |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| |   ∨ isrcv(kind(a(1of(e');2of(e'))))
| |     & e <c sender(e') ∨ e = sender(e')

| 15. e1 : Id
| 16. e2 : ℕ
| 17. ¬isnull(a(e1;e2))
```

```
| 18. isrcv(kind(a(e1;e2)))
| ⊢  isrcv(lnk(kind(a(e1;e2)));a(e1;e2))
| | by Unfold ‘w-isrcvl‘ 0 THEN AutoBoolCase isnull(a(e1;e2))
| |    THEN
| |    AutoBoolCase isrcv(kind(a(e1;e2)))
| |
| 1. the_w : World
| 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| 3. ∀i:Id, t:ℕ.
|    isnull(a(i;t)) ⇒ (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| 4. ∀i:Id, t:ℕ, l:IdLnk.
|    isrcv(l;a(i;t))
|    ⇒
|    destination(l) = i
|    & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| 5. ∀l:IdLnk, t:ℕ.
|    ∃t’:ℕ. t≤t’ & isrcv(l;a(destination(l);t’)) ∨ queue(l;t’) = nil
| 6. ∀e,e’:E. 1of(e) = 1of(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
| 7. ∀e:E. first(e) ⟺ (∀e’:E. 1of(e’) = 1of(e) ⇒ ¬e’ <c e)
| 8. ∀e:E.
|    ¬first(e)
|    ⇒
|    1of(pred(e)) = 1of(e) & pred(e) <c e
|    & (∀e’:E. 1of(e’) = 1of(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
| 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| 10. Trans e,e’:E. e <c e’
| 11. SWellFounded(e <c e’)
| 12. ∀e:E.
|     isrcv(kind(a(1of(e);2of(e))))
|     ⇒
|     sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
|     =
|     msg(lnk(kind(a(1of(e);2of(e))));tag(kind(a(1of(e);2of(
|                                                 e))));val(e))
| 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| 14. ∀e,e’:E.
|     e <c e’
|     ⇒
|     ¬first(e’) & e <c pred(e’) ∨ e = pred(e’)
|     ∨ isrcv(kind(a(1of(e’);2of(e’))))
|       & e <c sender(e’) ∨ e = sender(e’)
| 15. e1 : Id
| 16. e2 : ℕ
| 17. ¬isnull(a(e1;e2))
| 18. isrcv(kind(a(e1;e2)))
| 19. ¬isnull(a(e1;e2))
| 20. isrcv(kind(a(e1;e2)))
| ⊢  lnk(kind(a(e1;e2))) = lnk(kind(a(e1;e2)))
|   by RW assert_pushdownC 0
|
|\
| 1. the_w : World
| 2. FairFifo
| 3. ∀e,e’:E. loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
| 4. ∀e:E. first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
```

```
| 5. ∀e:E.
| |  ¬first(e)
| |  ⇒
| |  loc(pred(e)) = loc(e) & pred(e) <c e
| |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| 7. Trans e,e':E. e <c e'
| 8. SWellFounded(e <c e')
| 9. ∀e:E.
| |  isrcv(kind(e))
| |  ⇒
| |  sends(lnk(kind(e));sender(e))[index(e)]
| |  =
| |  msg(lnk(kind(e));tag(kind(e));val(e))
| 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| 11. ∀e,e':E.
| |   e <c e'
| |   ⇒
| |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| ⊢  ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | by Reduce 0 THEN Unfold `fair-fifo` 2 THEN ExRepD THEN Analyze -3
| |    THEN
| |    Analyze -4
| |    THEN
| |    All (Unfolds [`w-loc`;`w-time`;`w-ekind`])
| |    THEN
| |    All Reduce
| |    THEN
| |    All (Unfold `w-act`)
| |    THEN
| |    All Reduce
| |
| 1. the_w : World
| 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| 3. ∀i:Id, t:ℕ.
| |  isnull(a(i;t)) ⇒ (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| 4. ∀i:Id, t:ℕ, l:IdLnk.
| |  isrcv(l;a(i;t))
| |  ⇒
| |  destination(l) = i
| |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| 5. ∀l:IdLnk, t:ℕ.
| |  ∃t':ℕ. t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| 8. ∀e:E.
| |  ¬first(e)
| |  ⇒
| |  1of(pred(e)) = 1of(e) & pred(e) <c e
| |  & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| 10. Trans e,e':E. e <c e'
| 11. SWellFounded(e <c e')
```

```
| 12. ∀e:E.
| |   isrcv(kind(a(1of(e);2of(e))))
| |   ⇒
| |   sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| |   =
| |   msg(lnk(kind(a(1of(e);2of(e))));tag(kind(a(1of(e);2of(
| |                                                    e))));val(e))
| 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| 14. ∀e,e':E.
| |   e <c e'
| |   ⇒
| |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| |   ∨ isrcv(kind(a(1of(e');2of(e'))))
| |     & e <c sender(e') ∨ e = sender(e')
| 15. ∀e:E.
| |   isrcv(kind(a(1of(e);2of(e))))
| |   ⇒
| |   1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
| 16. e1 : Id
| 17. e2 : ℕ
| 18. ¬isnull(a(e1;e2))
| 19. l : IdLnk
| 20. ¬e1 = source(l)
| ⊢  sends(l;<e1,e2>) = nil  by InstHyp [e1;e2;l] 2
| |
| |\
| | .....antecedent .....
| | 1. the_w : World
| | 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| | 3. ∀i:Id, t:ℕ.
| |    isnull(a(i;t)) ⇒ (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| | 4. ∀i:Id, t:ℕ, l:IdLnk.
| |    isrcv(l;a(i;t))
| |    ⇒
| |    destination(l) = i
| |    & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| | 5. ∀l:IdLnk, t:ℕ.
| |    ∃t':ℕ. t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| | 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| | 8. ∀e:E.
| |    ¬first(e)
| |    ⇒
| |    1of(pred(e)) = 1of(e) & pred(e) <c e
| |    & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | 10. Trans e,e':E. e <c e'
| | 11. SWellFounded(e <c e')
| | 12. ∀e:E.
| |     isrcv(kind(a(1of(e);2of(e))))
| |     ⇒
| |     sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| |     =
| |     msg(lnk(kind(a(1of(e);2of(e))));tag(kind(a(1of(
| |                                                e);2of(
```

```
| |                                                e))));val(e))
| | 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| | 14. ∀e,e':E.
| |     e <c e'
| |     ⇒
| |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| |     ∨ isrcv(kind(a(1of(e');2of(e'))))
| |       & e <c sender(e') ∨ e = sender(e')
| | 15. ∀e:E.
| |     isrcv(kind(a(1of(e);2of(e))))
| |     ⇒
| |     1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
| | 16. e1 : Id
| | 17. e2 : ℕ
| | 18. ¬isnull(a(e1;e2))
| | 19. l : IdLnk
| | 20. ¬e1 = source(l)
| | ⊢  ¬source(l) = e1  by ParallelOp -1
| |
| .\
| . 1. the_w : World
| . 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| . 3. ∀i:Id, t:ℕ.
| . |  isnull(a(i;t)) ⇒ (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| . 4. ∀i:Id, t:ℕ, l:IdLnk.
| . |  isrcv(l;a(i;t))
| . |  ⇒
| . |  destination(l) = i
| . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| . 5. ∀l:IdLnk, t:ℕ.
| . |  ∃t':ℕ. t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| . 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| . 8. ∀e:E.
| . |  ¬first(e)
| . |  ⇒
| . |  1of(pred(e)) = 1of(e) & pred(e) <c e
| . |  & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . 10. Trans e,e':E. e <c e'
| . 11. SWellFounded(e <c e')
| . 12. ∀e:E.
| . |   isrcv(kind(a(1of(e);2of(e))))
| . |   ⇒
| . |   sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| . |   =
| . |   msg(lnk(kind(a(1of(e);2of(e))));tag(kind(a(1of(
| . |                                             e);2of(
| . |                                                e))));val(e))
| . 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| . 14. ∀e,e':E.
| . |   e <c e'
| . |   ⇒
| . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . |   ∨ isrcv(kind(a(1of(e');2of(e'))))
```

```
| . |     & e <c sender(e') ∨ e = sender(e')
| . 15. ∀e:E.
| . |   isrcv(kind(a(1of(e);2of(e))))
| . |   ⇒
| . |   1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
| . 16. e1 : Id
| . 17. e2 : ℕ
| . 18. ¬isnull(a(e1;e2))
| . 19. l : IdLnk
| . 20. ¬e1 = source(l)
| . 21. onlnk(l;m(e1;e2)) = nil
| . ⊢  sends(l;<e1,e2>) = nil
| . | by Unfold `w-sends` 0 THEN Unfolds [`w-loc`;`w-time`] 0
| . |    THEN
| . |    Reduce 0
| . |
| . 1. the_w : World
| . 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| . 3. ∀i:Id, t:ℕ.
| . |  isnull(a(i;t)) ⇒ (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| . 4. ∀i:Id, t:ℕ, l:IdLnk.
| . |  isrcv(l;a(i;t))
| . |  ⇒
| . |  destination(l) = i
| . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| . 5. ∀l:IdLnk, t:ℕ.
| . |  ∃t':ℕ. t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| . 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| . 8. ∀e:E.
| . |  ¬first(e)
| . |  ⇒
| . |  1of(pred(e)) = 1of(e) & pred(e) <c e
| . |  & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . 10. Trans e,e':E. e <c e'
| . 11. SWellFounded(e <c e')
| . 12. ∀e:E.
| . |   isrcv(kind(a(1of(e);2of(e))))
| . |   ⇒
| . |   sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| . |   =
| . |   msg(lnk(kind(a(1of(e);2of(e))));tag(kind(a(1of(
| . |                                              e);2of(
| . |                                                 e))));val(e))
| . 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| . 14. ∀e,e':E.
| . |   e <c e'
| . |   ⇒
| . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . |   ∨ isrcv(kind(a(1of(e');2of(e'))))
| . |     & e <c sender(e') ∨ e = sender(e')
| . 15. ∀e:E.
| . |   isrcv(kind(a(1of(e);2of(e))))
| . |   ⇒
```

```
| . |   1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
| . 16. e1 : Id
| . 17. e2 : ℕ
| . 18. ¬isnull(a(e1;e2))
| . 19. l : IdLnk
| . 20. ¬e1 = source(l)
| . 21. onlnk(l;m(e1;e2)) = nil
| . ⊢  onlnk(l;m(e1;e2)) = nil  by RWO "assert_of_null<" -1
| . |
| . 1. the_w : World
| . 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| . 3. ∀i:Id, t:ℕ.
| . |  isnull(a(i;t)) ⇒ (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| . 4. ∀i:Id, t:ℕ, l:IdLnk.
| . |  isrcv(l;a(i;t))
| . |  ⇒
| . |  destination(l) = i
| . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| . 5. ∀l:IdLnk, t:ℕ.
| . |  ∃t':ℕ. t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| . 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| . 8. ∀e:E.
| . |  ¬first(e)
| . |  ⇒
| . |  1of(pred(e)) = 1of(e) & pred(e) <c e
| . |  & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . 10. Trans e,e':E. e <c e'
| . 11. SWellFounded(e <c e')
| . 12. ∀e:E.
| . |   isrcv(kind(a(1of(e);2of(e))))
| . |   ⇒
| . |   sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| . |   =
| . |   msg(lnk(kind(a(1of(e);2of(e))));tag(kind(a(1of(
| . |                                           e);2of(
| . |                                              e))));val(e))
| . 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| . 14. ∀e,e':E.
| . |   e <c e'
| . |   ⇒
| . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . |   ∨ isrcv(kind(a(1of(e');2of(e'))))
| . |     & e <c sender(e') ∨ e = sender(e')
| . 15. ∀e:E.
| . |   isrcv(kind(a(1of(e);2of(e))))
| . |   ⇒
| . |   1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
| . 16. e1 : Id
| . 17. e2 : ℕ
| . 18. ¬isnull(a(e1;e2))
| . 19. l : IdLnk
| . 20. ¬e1 = source(l)
| . 21. null(onlnk(l;m(e1;e2)))
```

```
| . ⊢  onlnk(l;m(e1;e2)) = nil
| . | by RWO "assert_of_null<" 0 THEN Try Trivial
| . |
| . |\
| . | .....wf .....
| . | 1. the_w : World
| . | 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| . | 3. ∀i:Id, t:ℕ.
| . |    isnull(a(i;t))
| . |    ⇒
| . |    (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| . | 4. ∀i:Id, t:ℕ, l:IdLnk.
| . |    isrcv(l;a(i;t))
| . |    ⇒
| . |    destination(l) = i
| . |    & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| . | 5. ∀l:IdLnk, t:ℕ.
| . |    ∃t':ℕ.
| . |    t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| . | 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| . | 8. ∀e:E.
| . |    ¬first(e)
| . |    ⇒
| . |    1of(pred(e)) = 1of(e) & pred(e) <c e
| . |    & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | 10. Trans e,e':E. e <c e'
| . | 11. SWellFounded(e <c e')
| . | 12. ∀e:E.
| . |     isrcv(kind(a(1of(e);2of(e))))
| . |     ⇒
| . |     sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| . |     =
| . |     msg(lnk(kind(a(1of(e);2of(e))));tag(kind(a(1of(
| . |                                             e);2of(
| . |                                               e))));val(e))
| . | 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| . | 14. ∀e,e':E.
| . |     e <c e'
| . |     ⇒
| . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . |     ∨ isrcv(kind(a(1of(e');2of(e'))))
| . |       & e <c sender(e') ∨ e = sender(e')
| . | 15. ∀e:E.
| . |     isrcv(kind(a(1of(e);2of(e))))
| . |     ⇒
| . |     1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
| . | 16. e1 : Id
| . | 17. e2 : ℕ
| . | 18. ¬isnull(a(e1;e2))
| . | 19. l : IdLnk
| . | 20. ¬e1 = source(l)
| . | 21. null(onlnk(l;m(e1;e2)))
| . | ⊢  Msg_sub(l; the_w.M) ∈ Type  by Auto
```

```
| . |
| . .\
| . . .....wf .....
| . . 1. the_w : World
| . . 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| . . 3. ∀i:Id, t:ℕ.
| . . |  isnull(a(i;t))
| . . |  ⇒
| . . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| . . 4. ∀i:Id, t:ℕ, l:IdLnk.
| . . |  isrcv(l;a(i;t))
| . . |  ⇒
| . . |  destination(l) = i
| . . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| . . 5. ∀l:IdLnk, t:ℕ.
| . . |  ∃t':ℕ.
| . . |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| . . 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . . 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| . . 8. ∀e:E.
| . . |  ¬first(e)
| . . |  ⇒
| . . |  1of(pred(e)) = 1of(e) & pred(e) <c e
| . . |  & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . . 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . . 10. Trans e,e':E. e <c e'
| . . 11. SWellFounded(e <c e')
| . . 12. ∀e:E.
| . . |   isrcv(kind(a(1of(e);2of(e))))
| . . |   ⇒
| . . |   sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| . . |   =
| . . |   msg(lnk(kind(a(1of(e);2of(e))));tag(kind(a(1of(
| . . |                                             e);2of(
| . . |                                               e))));val(e))
| . . 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| . . 14. ∀e,e':E.
| . . |   e <c e'
| . . |   ⇒
| . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . . |   ∨ isrcv(kind(a(1of(e');2of(e'))))
| . . |     & e <c sender(e') ∨ e = sender(e')
| . . 15. ∀e:E.
| . . |   isrcv(kind(a(1of(e);2of(e))))
| . . |   ⇒
| . . |   1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
| . . 16. e1 : Id
| . . 17. e2 : ℕ
| . . 18. ¬isnull(a(e1;e2))
| . . 19. l : IdLnk
| . . 20. ¬e1 = source(l)
| . . 21. null(onlnk(l;m(e1;e2)))
| . . ⊢  onlnk(l;m(e1;e2)) ∈ Msg_sub(l; the_w.M) List
| . . | by Unfolds [`Msg_sub`;`w-onlnk`] 0
| . . |
```

```
| . . 1. the_w : World
| . . 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| . . 3. ∀i:Id, t:ℕ.
| . . |  isnull(a(i;t))
| . . |  ⇒
| . . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| . . 4. ∀i:Id, t:ℕ, l:IdLnk.
| . . |  isrcv(l;a(i;t))
| . . |  ⇒
| . . |  destination(l) = i
| . . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| . . 5. ∀l:IdLnk, t:ℕ.
| . . |  ∃t':ℕ.
| . . |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| . . 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . . 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| . . 8. ∀e:E.
| . . |  ¬first(e)
| . . |  ⇒
| . . |  1of(pred(e)) = 1of(e) & pred(e) <c e
| . . |  & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . . 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . . 10. Trans e,e':E. e <c e'
| . . 11. SWellFounded(e <c e')
| . . 12. ∀e:E.
| . . |   isrcv(kind(a(1of(e);2of(e))))
| . . |   ⇒
| . . |   sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| . . |   =
| . . |   msg(lnk(kind(a(1of(e);2of(e))));tag(kind(a(1of(
| . . |                                                 e);2of(
| . . |                                                    e))));val(e))
| . . 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| . . 14. ∀e,e':E.
| . . |   e <c e'
| . . |   ⇒
| . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . . |   ∨ isrcv(kind(a(1of(e');2of(e'))))
| . . |     & e <c sender(e') ∨ e = sender(e')
| . . 15. ∀e:E.
| . . |   isrcv(kind(a(1of(e);2of(e))))
| . . |   ⇒
| . . |   1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
| . . 16. e1 : Id
| . . 17. e2 : ℕ
| . . 18. ¬isnull(a(e1;e2))
| . . 19. l : IdLnk
| . . 20. ¬e1 = source(l)
| . . 21. null(onlnk(l;m(e1;e2)))
| . . ⊢  filter(λms.mlnk(ms) = l;m(e1;e2))
| . . |  ∈ {m:Msg(the_w.M)| haslink(l; m) } List
| . . | by GenConcl (filter(λms.mlnk(ms) = l;m(e1;e2)) = LS)
| . . |
| . . |\
| . . | .....wf .....
```

```
| . . | 1. the_w : World
| . . | 2. ∀i:Id, t:ℕ, l:IdLnk.
| . . |    ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| . . | 3. ∀i:Id, t:ℕ.
| . . |    isnull(a(i;t))
| . . |    ⇒
| . . |    (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| . . | 4. ∀i:Id, t:ℕ, l:IdLnk.
| . . |    isrcv(l;a(i;t))
| . . |    ⇒
| . . |    destination(l) = i
| . . |    & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| . . | 5. ∀l:IdLnk, t:ℕ.
| . . |    ∃t':ℕ.
| . . |    t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| . . | 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . . | 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| . . | 8. ∀e:E.
| . . |    ¬first(e)
| . . |    ⇒
| . . |    1of(pred(e)) = 1of(e) & pred(e) <c e
| . . |    & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . . | 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . . | 10. Trans e,e':E. e <c e'
| . . | 11. SWellFounded(e <c e')
| . . | 12. ∀e:E.
| . . |     isrcv(kind(a(1of(e);2of(e))))
| . . |     ⇒
| . . |     sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| . . |     =
| . . |     msg(lnk(kind(a(1of(
| . . |                    e);2of(e))));tag(kind(a(1of(
| . . |                                                e);2of(
| . . |                                                  e))));val(e))
| . . | 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| . . | 14. ∀e,e':E.
| . . |     e <c e'
| . . |     ⇒
| . . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . . |     ∨ isrcv(kind(a(1of(e');2of(e'))))
| . . |       & e <c sender(e') ∨ e = sender(e')
| . . | 15. ∀e:E.
| . . |     isrcv(kind(a(1of(e);2of(e))))
| . . |     ⇒
| . . |     1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
| . . | 16. e1 : Id
| . . | 17. e2 : ℕ
| . . | 18. ¬isnull(a(e1;e2))
| . . | 19. l : IdLnk
| . . | 20. ¬e1 = source(l)
| . . | 21. null(onlnk(l;m(e1;e2)))
| . . | ⊢  filter(λms.mlnk(ms) = l;m(e1;e2))
| . . |    ∈ {x:Msg(the_w.M)| mlnk(x) = l } List
| . . |   by Inst
| . . |      Thm* ∀T:Type, P:(T→𝔹), l:T List.
```

```
| . . |           filter(P;l) ∈ {x:T| P(x) } List
| . . |      {filter_type}
| . . |      [Msg(the_w.M);λms.mlnk(ms) = l;m(e1;e2)]
| . . |      THEN
| . . |      Reduce -1
| . . |      THEN
| . . |      Unfold `w-Msg` -1
| . . |
| . . |\
| . . | .....wf .....
| . . | 1. the_w : World
| . . | 2. ∀i:Id, t:ℕ, l:IdLnk.
| . . |    ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| . . | 3. ∀i:Id, t:ℕ.
| . . |    isnull(a(i;t))
| . . |    ⇒
| . . |    (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| . . | 4. ∀i:Id, t:ℕ, l:IdLnk.
| . . |    isrcv(l;a(i;t))
| . . |    ⇒
| . . |    destination(l) = i
| . . |    & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| . . | 5. ∀l:IdLnk, t:ℕ.
| . . |    ∃t':ℕ.
| . . |    t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| . . | 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . . | 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| . . | 8. ∀e:E.
| . . |    ¬first(e)
| . . |    ⇒
| . . |    1of(pred(e)) = 1of(e) & pred(e) <c e
| . . |    & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . . | 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . . | 10. Trans e,e':E. e <c e'
| . . | 11. SWellFounded(e <c e')
| . . | 12. ∀e:E.
| . . |     isrcv(kind(a(1of(e);2of(e))))
| . . |     ⇒
| . . |     sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| . . |     =
| . . |     msg(lnk(kind(a(1of(
| . . |                    e);2of(e))));tag(kind(a(1of(
| . . |                                              e);2of(
| . . |                                                 e))));val(e))
| . . | 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| . . | 14. ∀e,e':E.
| . . |     e <c e'
| . . |     ⇒
| . . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . . |     ∨ isrcv(kind(a(1of(e');2of(e'))))
| . . |       & e <c sender(e') ∨ e = sender(e')
| . . | 15. ∀e:E.
| . . |     isrcv(kind(a(1of(e);2of(e))))
| . . |     ⇒
| . . |     1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
```

```
| . . | 16. e1 : Id
| . . | 17. e2 : ℕ
| . . | 18. ¬isnull(a(e1;e2))
| . . | 19. l : IdLnk
| . . | 20. ¬e1 = source(l)
| . . | 21. null(onlnk(l;m(e1;e2)))
| . . | ⊢  {x:Msg(the_w.M)| mlnk(x) = l } List ∈ Type  by Auto
| . . |
| . . .\
| . . . 1. the_w : World
| . . . 2. ∀i:Id, t:ℕ, l:IdLnk.
| . . . |  ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| . . . 3. ∀i:Id, t:ℕ.
| . . . |  isnull(a(i;t))
| . . . |  ⇒
| . . . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| . . . 4. ∀i:Id, t:ℕ, l:IdLnk.
| . . . |  isrcv(l;a(i;t))
| . . . |  ⇒
| . . . |  destination(l) = i
| . . . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| . . . 5. ∀l:IdLnk, t:ℕ.
| . . . |  ∃t':ℕ.
| . . . |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| . . . 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . . . 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| . . . 8. ∀e:E.
| . . . |  ¬first(e)
| . . . |  ⇒
| . . . |  1of(pred(e)) = 1of(e) & pred(e) <c e
| . . . |  & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . . . 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . . . 10. Trans e,e':E. e <c e'
| . . . 11. SWellFounded(e <c e')
| . . . 12. ∀e:E.
| . . . |   isrcv(kind(a(1of(e);2of(e))))
| . . . |   ⇒
| . . . |   sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| . . . |   =
| . . . |   msg(lnk(kind(a(1of(
| . . . |                  e);2of(e))));tag(kind(a(1of(
| . . . |                                              e);2of(
| . . . |                                                 e))));val(e))
| . . . 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| . . . 14. ∀e,e':E.
| . . . |   e <c e'
| . . . |   ⇒
| . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . . . |   ∨ isrcv(kind(a(1of(e');2of(e'))))
| . . . |     & e <c sender(e') ∨ e = sender(e')
| . . . 15. ∀e:E.
| . . . |   isrcv(kind(a(1of(e);2of(e))))
| . . . |   ⇒
| . . . |   1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
| . . . 16. e1 : Id
```

```
| . . . 17. e2 : ℕ
| . . . 18. ¬isnull(a(e1;e2))
| . . . 19. l : IdLnk
| . . . 20. ¬e1 = source(l)
| . . . 21. null(onlnk(l;m(e1;e2)))
| . . . 22. LS : {x:Msg(the_w.M)| mlnk(x) = l } List
| . . . 23. filter(λms.mlnk(ms) = l;m(e1;e2)) = LS
| . . . ⊢  LS ∈ {m:Msg(the_w.M)| haslink(l; m) } List
| . . . | by Thin -1 THEN MoveToConcl -1 THEN InductionOnList
| . . . |
| . . . .....subterm .....
| . . . 1. the_w : World
| . . . 2. ∀i:Id, t:ℕ, l:IdLnk.
| . . . |  ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| . . . 3. ∀i:Id, t:ℕ.
| . . . |  isnull(a(i;t))
| . . . |  ⇒
| . . . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| . . . 4. ∀i:Id, t:ℕ, l:IdLnk.
| . . . |  isrcv(l;a(i;t))
| . . . |  ⇒
| . . . |  destination(l) = i
| . . . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| . . . 5. ∀l:IdLnk, t:ℕ.
| . . . |  ∃t':ℕ.
| . . . |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| . . . 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . . . 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| . . . 8. ∀e:E.
| . . . |  ¬first(e)
| . . . |  ⇒
| . . . |  1of(pred(e)) = 1of(e) & pred(e) <c e
| . . . |  & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . . . 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . . . 10. Trans e,e':E. e <c e'
| . . . 11. SWellFounded(e <c e')
| . . . 12. ∀e:E.
| . . . |   isrcv(kind(a(1of(e);2of(e))))
| . . . |   ⇒
| . . . |   sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| . . . |   =
| . . . |   msg(lnk(kind(a(1of(
| . . . |                   e);2of(e))));tag(kind(a(1of(
| . . . |                                           e);2of(
| . . . |                                              e))));val(e))
| . . . 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| . . . 14. ∀e,e':E.
| . . . |   e <c e'
| . . . |   ⇒
| . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . . . |   ∨ isrcv(kind(a(1of(e');2of(e'))))
| . . . |     & e <c sender(e') ∨ e = sender(e')
| . . . 15. ∀e:E.
| . . . |   isrcv(kind(a(1of(e);2of(e))))
| . . . |   ⇒
```

```
| . . . |   1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
| . . . 16. e1 : Id
| . . . 17. e2 : ℕ
| . . . 18. ¬isnull(a(e1;e2))
| . . . 19. l : IdLnk
| . . . 20. ¬e1 = source(l)
| . . . 21. null(onlnk(l;m(e1;e2)))
| . . . 22. {x:Msg(the_w.M)| mlnk(x) = l } List
| . . . 23. u : {x:Msg(the_w.M)| mlnk(x) = l }
| . . . 24. v : {x:Msg(the_w.M)| mlnk(x) = l } List
| . . . 25. v ∈ {m:Msg(the_w.M)| haslink(l; m) } List
| . . . ⊢  u ∈ {m:Msg(the_w.M)| haslink(l; m) }
| . . . | by Analyze -3 THEN Analyze
| . . . |
| . . . .....set predicate .....
| . . . 1. the_w : World
| . . . 2. ∀i:Id, t:ℕ, l:IdLnk.
| . . .    ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
| . . . 3. ∀i:Id, t:ℕ.
| . . .    isnull(a(i;t))
| . . .    ⇒
| . . .    (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
| . . . 4. ∀i:Id, t:ℕ, l:IdLnk.
| . . .    isrcv(l;a(i;t))
| . . .    ⇒
| . . .    destination(l) = i
| . . .    & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
| . . . 5. ∀l:IdLnk, t:ℕ.
| . . .    ∃t':ℕ.
| . . .    t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
| . . . 6. ∀e,e':E. 1of(e) = 1of(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . . . 7. ∀e:E. first(e) ⟺ (∀e':E. 1of(e') = 1of(e) ⇒ ¬e' <c e)
| . . . 8. ∀e:E.
| . . .    ¬first(e)
| . . .    ⇒
| . . .    1of(pred(e)) = 1of(e) & pred(e) <c e
| . . .    & (∀e':E. 1of(e') = 1of(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . . . 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . . . 10. Trans e,e':E. e <c e'
| . . . 11. SWellFounded(e <c e')
| . . . 12. ∀e:E.
| . . .     isrcv(kind(a(1of(e);2of(e))))
| . . .     ⇒
| . . .     sends(lnk(kind(a(1of(e);2of(e))));sender(e))[index(e)]
| . . .     =
| . . .     msg(lnk(kind(a(1of(
| . . .                    e);2of(e))));tag(kind(a(1of(
| . . .                                             e);2of(
| . . .                                                e))));val(e))
| . . . 13. ∀e:E. isrcv(kind(a(1of(e);2of(e)))) ⇒ sender(e) <c e
| . . . 14. ∀e,e':E.
| . . .     e <c e'
| . . .     ⇒
| . . .     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . . .     ∨ isrcv(kind(a(1of(e');2of(e'))))
```

```
| . . .       & e <c sender(e') ∨ e = sender(e')
| . . . 15. ∀e:E.
| . . .     isrcv(kind(a(1of(e);2of(e))))
| . . .     ⇒
| . . .     1of(e) = destination(lnk(kind(a(1of(e);2of(e)))))
| . . . 16. e1 : Id
| . . . 17. e2 : ℕ
| . . . 18. ¬isnull(a(e1;e2))
| . . . 19. l : IdLnk
| . . . 20. ¬e1 = source(l)
| . . . 21. null(onlnk(l;m(e1;e2)))
| . . . 22. {x:Msg(the_w.M)| mlnk(x) = l } List
| . . . 23. u : Msg(the_w.M)
| . . . 24. mlnk(u) = l
| . . . 25. v : {x:Msg(the_w.M)| mlnk(x) = l } List
| . . . 26. v ∈ {m:Msg(the_w.M)| haslink(l; m) } List
| . . . ⊢  haslink(l; u)
| . . .   by Unfold `haslink` 0 THEN RW assert_pushdownC -3
|
|\
| 1. the_w : World
| 2. FairFifo
| 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| 5. ∀e:E.
| |  ¬first(e)
| |  ⇒
| |  loc(pred(e)) = loc(e) & pred(e) <c e
| |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| 7. Trans e,e':E. e <c e'
| 8. SWellFounded(e <c e')
| 9. ∀e:E.
| |  isrcv(kind(e))
| |  ⇒
| |  sends(lnk(kind(e));sender(e))[index(e)]
| |  =
| |  msg(lnk(kind(e));tag(kind(e));val(e))
| 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| 11. ∀e,e':E.
| |   e <c e'
| |   ⇒
| |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| ⊢  ∀e,e':E.
| |  isrcv(kind(e))
| |  ⇒
| |  isrcv(kind(e'))
| |  ⇒
| |  lnk(kind(e)) = lnk(kind(e'))
| |  ⇒
| |  (e <c e'
| |   ⟺
```

```
| |   sender(e) <c sender(e')
| |   ∨ sender(e) = sender(e') & index(e)<index(e'))
| | by Assert
| |     (∀e,e':E.
| |      isrcv(kind(e))
| |      ⇒
| |      isrcv(kind(e'))
| |      ⇒
| |      lnk(kind(e)) = lnk(kind(e'))
| |      ⇒
| |      (e <loc e'
| |       ⟺
| |       sender(e) <loc sender(e')
| |       ∨ sender(e) = sender(e') & index(e)<index(e')))
| |
| |\
| | .....assertion .....
| | 1. the_w : World
| | 2. FairFifo
| | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | 5. ∀e:E.
| | |  ¬first(e)
| | |  ⇒
| | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | 7. Trans e,e':E. e <c e'
| | 8. SWellFounded(e <c e')
| | 9. ∀e:E.
| | |  isrcv(kind(e))
| | |  ⇒
| | |  sends(lnk(kind(e));sender(e))[index(e)]
| | |  =
| | |  msg(lnk(kind(e));tag(kind(e));val(e))
| | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | 11. ∀e,e':E.
| | |   e <c e'
| | |   ⇒
| | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | ⊢  ∀e,e':E.
| | |  isrcv(kind(e))
| | |  ⇒
| | |  isrcv(kind(e'))
| | |  ⇒
| | |  lnk(kind(e)) = lnk(kind(e'))
| | |  ⇒
| | |  (e <loc e'
| | |   ⟺
| | |   sender(e) <loc sender(e')
| | |   ∨ sender(e) = sender(e') & index(e)<index(e'))
| | | by RepeatFor 5 (Analyze 0)
```

```
| | |
| | 1. the_w : World
| | 2. FairFifo
| | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | 5. ∀e:E.
| | |  ¬first(e)
| | |  ⇒
| | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | 7. Trans e,e':E. e <c e'
| | 8. SWellFounded(e <c e')
| | 9. ∀e:E.
| | |  isrcv(kind(e))
| | |  ⇒
| | |  sends(lnk(kind(e));sender(e))[index(e)]
| | |  =
| | |  msg(lnk(kind(e));tag(kind(e));val(e))
| | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | 11. ∀e,e':E.
| | |   e <c e'
| | |   ⇒
| | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | 14. e : E
| | 15. e' : E
| | 16. isrcv(kind(e))
| | 17. isrcv(kind(e'))
| | 18. lnk(kind(e)) = lnk(kind(e'))
| | ⊢  e <loc e'
| | |  ⟺
| | |  sender(e) <loc sender(e')
| | |  ∨ sender(e) = sender(e') & index(e)<index(e')
| | | by RWO "assert-w-eq-E-iff<" 0
| | |
| | 1. the_w : World
| | 2. FairFifo
| | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | 5. ∀e:E.
| | |  ¬first(e)
| | |  ⇒
| | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | 7. Trans e,e':E. e <c e'
| | 8. SWellFounded(e <c e')
| | 9. ∀e:E.
| | |  isrcv(kind(e))
| | |  ⇒
| | |  sends(lnk(kind(e));sender(e))[index(e)]
| | |  =
```

```
| | |  msg(lnk(kind(e));tag(kind(e));val(e))
| | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | 11. ∀e,e':E.
| | |   e <c e'
| | |   ⇒
| | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | 14. e : E
| | 15. e' : E
| | 16. isrcv(kind(e))
| | 17. isrcv(kind(e'))
| | 18. lnk(kind(e)) = lnk(kind(e'))
| | ⊢  e <loc e'
| | |  ⟺
| | |  sender(e) <loc sender(e')
| | |  ∨ sender(e) = sender(e') & index(e)<index(e')
| | | by Unfold `w-index` 0 THEN Unfold `w-sender` 0
| | |    THEN
| | |    Assert
| | |    match(lnk(kind(e));mu(λt.
| | |    match(lnk(kind(e));t;time(e)));time(e))
| | |
| | |\
| | | .....assertion .....
| | | 1. the_w : World
| | | 2. FairFifo
| | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | | 5. ∀e:E.
| | |    ¬first(e)
| | |    ⇒
| | |    loc(pred(e)) = loc(e) & pred(e) <c e
| | |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | | 7. Trans e,e':E. e <c e'
| | | 8. SWellFounded(e <c e')
| | | 9. ∀e:E.
| | |    isrcv(kind(e))
| | |    ⇒
| | |    sends(lnk(kind(e));sender(e))[index(e)]
| | |    =
| | |    msg(lnk(kind(e));tag(kind(e));val(e))
| | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | | 11. ∀e,e':E.
| | |     e <c e'
| | |     ⇒
| | |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | | 14. e : E
| | | 15. e' : E
| | | 16. isrcv(kind(e))
```

```
| | | 17. isrcv(kind(e'))
| | | 18. lnk(kind(e)) = lnk(kind(e'))
| | | ⊢  match(lnk(kind(e));mu(λt.
| | |    match(lnk(kind(e));t;time(e)));time(e))
| | |   by AssertBY (∃n:ℕtime(e). match(lnk(kind(e));n;time(e)))
| | |      (BackThru
| | |       Thm* ∀the_w:World, e:E.
| | |             FairFifo
| | |             ⇒
| | |             isrcv(kind(e))
| | |             ⇒
| | |             (∃t:ℕtime(e). match(lnk(kind(e));t;time(e)))
| | |       {w_DASH_match_DASH_exists})
| | |      THEN
| | |      Inst
| | |      Thm* ∀f:(ℕ→𝔹).
| | |           (∃n:ℕ. f(n)) ⇒ f(mu(f)) & (∀i:ℕ. i<mu(f) ⇒ ¬f(i))
| | |      {mu_DASH_property}
| | |      [λt.match(lnk(kind(e));t;time(e))]
| | |      THENA
| | |      (Reduce 0 THEN ParallelOp -1)
| | |      THEN
| | |      Reduce -1
| | |
| | .\
| | . 1. the_w : World
| | . 2. FairFifo
| | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . 5. ∀e:E.
| | . |  ¬first(e)
| | . |  ⇒
| | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . 7. Trans e,e':E. e <c e'
| | . 8. SWellFounded(e <c e')
| | . 9. ∀e:E.
| | . |  isrcv(kind(e))
| | . |  ⇒
| | . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . |  =
| | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . 11. ∀e,e':E.
| | . |   e <c e'
| | . |   ⇒
| | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . 14. e : E
| | . 15. e' : E
| | . 16. isrcv(kind(e))
| | . 17. isrcv(kind(e'))
```

```
| | . 18. lnk(kind(e)) = lnk(kind(e'))
| | . 19. match(lnk(kind(e));mu(λt.
| | . |    match(lnk(kind(e));t;time(e)));time(e))
| | . ⊢  e <loc e'
| | . |  ⟺
| | . |  <source(lnk(kind(e)))
| | . |  ,mu(λt.
| | . |   match(lnk(kind(e));t;time(e)))> <loc ...
| | . |  ∨ <source(lnk(kind(e)))
| | . |    ,mu(λt.
| | . |     match(lnk(kind(e));t;time(e)))> = <source(lnk(kind(e')))
| | . |                                        ,mu(λt.
| | . |                                         match(lnk(...);t;...))>
| | . |    & ||rcvs(lnk(kind(e));time(e))||
| | . |      -||snds(lnk(kind(e));time(...))||<||rcvs(lnk(...);...)||
| | . |      -||snds(lnk(kind(e'));time(...))||
| | . | by Assert
| | . |    match(lnk(kind(e'));mu(λt.
| | . |    match(lnk(kind(e'));t;time(e')));time(e'))
| | . |
| | . |\
| | . | .....assertion .....
| | . | 1. the_w : World
| | . | 2. FairFifo
| | . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . | 5. ∀e:E.
| | . |    ¬first(e)
| | . |    ⇒
| | . |    loc(pred(e)) = loc(e) & pred(e) <c e
| | . |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . | 7. Trans e,e':E. e <c e'
| | . | 8. SWellFounded(e <c e')
| | . | 9. ∀e:E.
| | . |    isrcv(kind(e))
| | . |    ⇒
| | . |    sends(lnk(kind(e));sender(e))[index(e)]
| | . |    =
| | . |    msg(lnk(kind(e));tag(kind(e));val(e))
| | . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . | 11. ∀e,e':E.
| | . |     e <c e'
| | . |     ⇒
| | . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . | 14. e : E
| | . | 15. e' : E
| | . | 16. isrcv(kind(e))
| | . | 17. isrcv(kind(e'))
| | . | 18. lnk(kind(e)) = lnk(kind(e'))
| | . | 19. match(lnk(kind(e));mu(λt.
| | . |     match(lnk(kind(e));t;time(e)));time(e))
```

```
| | . | ⊢  match(lnk(kind(e'));mu(λt.
| | . |    match(lnk(kind(e'));t;time(e')));time(e'))
| | . |   by AssertBY (∃n:ℕtime(e'). match(lnk(kind(e'));n;time(e')))
| | . |      (BackThru
| | . |       Thm* ∀the_w:World, e:E.
| | . |            FairFifo
| | . |            ⇒
| | . |            isrcv(kind(e))
| | . |            ⇒
| | . |            (∃t:ℕtime(e). match(lnk(kind(e));t;time(e)))
| | . |       {w_DASH_match_DASH_exists})
| | . |      THEN
| | . |      Inst
| | . |      Thm* ∀f:(ℕ→𝔹).
| | . |           (∃n:ℕ. f(n)) ⇒ f(mu(f)) & (∀i:ℕ. i<mu(f) ⇒ ¬f(i))
| | . |      {mu_DASH_property}
| | . |      [λt.match(lnk(kind(e'));t;time(e'))]
| | . |      THENA
| | . |      (Reduce 0 THEN ParallelOp -1)
| | . |      THEN
| | . |      Reduce -1
| | . |
| | . .\
| | . . 1. the_w : World
| | . . 2. FairFifo
| | . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . 5. ∀e:E.
| | . . |  ¬first(e)
| | . . |  ⇒
| | . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . 7. Trans e,e':E. e <c e'
| | . . 8. SWellFounded(e <c e')
| | . . 9. ∀e:E.
| | . . |  isrcv(kind(e))
| | . . |  ⇒
| | . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . |  =
| | . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . 11. ∀e,e':E.
| | . . |   e <c e'
| | . . |   ⇒
| | . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . 14. e : E
| | . . 15. e' : E
| | . . 16. isrcv(kind(e))
| | . . 17. isrcv(kind(e'))
| | . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . 19. match(lnk(kind(e));mu(λt.
```

```
| | . . |   match(lnk(kind(e));t;time(e)));time(e))
| | . . 20. match(lnk(kind(e'));mu(λt.
| | . . |   match(lnk(kind(e'));t;time(e')));time(e'))
| | . . ⊢  e <loc e'
| | . . |  ⟺
| | . . |  <source(lnk(kind(e)))
| | . . |  ,mu(λt.
| | . . |   match(lnk(kind(e));t;time(e)))> <loc <source(lnk(...))
| | . . |                                        ,mu(λt.
| | . . |                                         match(...;t;...))>
| | . . |  ∨ <source(lnk(kind(e)))
| | . . |    ,mu(λt.
| | . . |     match(lnk(kind(e));t;time(e)))> = <source(lnk(kind(e')))
| | . . |                                       ,mu(λt.
| | . . |                                        match(...;t;...))>
| | . . |    & ||rcvs(lnk(kind(e));time(e))||
| | . . |      -||snds(lnk(kind(e));time(...))||<||rcvs(...;...)||
| | . . |      -||snds(lnk(kind(e'));time(...))||
| | . . | by RepeatFor 2 (MoveToConcl -1)
| | . . |    THEN
| | . . |    GenConcl (mu(λt.match(lnk(kind(e'));t;time(e'))) = t')
| | . . |
| | . . |\
| | . . | 1. the_w : World
| | . . | 2. FairFifo
| | . . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . | 5. ∀e:E.
| | . . |    ¬first(e)
| | . . |    ⇒
| | . . |    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . | 6. ∀e:E.
| | . . |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . | 7. Trans e,e':E. e <c e'
| | . . | 8. SWellFounded(e <c e')
| | . . | 9. ∀e:E.
| | . . |    isrcv(kind(e))
| | . . |    ⇒
| | . . |    sends(lnk(kind(e));sender(e))[index(e)]
| | . . |    =
| | . . |    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . | 11. ∀e,e':E.
| | . . |     e <c e'
| | . . |     ⇒
| | . . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . | 12. ∀e:E.
| | . . |     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . | 14. e : E
| | . . | 15. e' : E
| | . . | 16. isrcv(kind(e))
| | . . | 17. isrcv(kind(e'))
```

```
| | . . | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . | ⊢  ∃n:ℕ. (λt.match(lnk(kind(e'));t;time(e')))(n)
| | . . |   by Reduce 0
| | . . |      THEN
| | . . |      Inst
| | . . |      Thm* ∀the_w:World, e:E.
| | . . |           FairFifo
| | . . |           ⇒
| | . . |           isrcv(kind(e))
| | . . |           ⇒
| | . . |           (∃t:ℕtime(e). match(lnk(kind(e));t;time(e)))
| | . . |      {w_DASH_match_DASH_exists}
| | . . |      [the_w;e']
| | . . |      THEN
| | . . |      ParallelOp -1
| | . . |
| | . . .\
| | . . . 1. the_w : World
| | . . . 2. FairFifo
| | . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . 5. ∀e:E.
| | . . . |  ¬first(e)
| | . . . |  ⇒
| | . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . 6. ∀e:E.
| | . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . . 7. Trans e,e':E. e <c e'
| | . . . 8. SWellFounded(e <c e')
| | . . . 9. ∀e:E.
| | . . . |  isrcv(kind(e))
| | . . . |  ⇒
| | . . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . |  =
| | . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . 11. ∀e,e':E.
| | . . . |   e <c e'
| | . . . |   ⇒
| | . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . . 12. ∀e:E.
| | . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . 14. e : E
| | . . . 15. e' : E
| | . . . 16. isrcv(kind(e))
| | . . . 17. isrcv(kind(e'))
| | . . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . 19. t' : ℕ
| | . . . 20. mu(λt.match(lnk(kind(e'));t;time(e'))) = t'
| | . . . ⊢  match(lnk(kind(e));mu(λt.
| | . . . |  match(lnk(kind(e));t;time(e)));time(e))
| | . . . |  ⇒
```

```
| | . . . |  match(lnk(kind(e'));t';time(e'))
| | . . . |  ⇒
| | . . . |  (e <loc e'
| | . . . |   ⟺
| | . . . |   <source(lnk(kind(e)))
| | . . . |   ,mu(λt.
| | . . . |    match(lnk(kind(e));t;time(e)))> <loc <source(lnk(...))
| | . . . |                                          ,t'>
| | . . . |   ∨ <source(lnk(kind(e)))
| | . . . |      ,mu(λt.
| | . . . |       match(lnk(kind(e));t;time(e)))> = <source(lnk(...))
| | . . . |                                         ,t'>
| | . . . |      & ||rcvs(lnk(kind(e));time(e))||
| | . . . |        -||snds(lnk(kind(e));time(...))||<||rcvs(...;...)||
| | . . . |        -||snds(lnk(kind(e'));time(<source(lnk(kind(e')))
| | . . . |                                  ,t'>))||)
| | . . . | by GenConcl (mu(λt.match(lnk(kind(e));t;time(e))) = t)
| | . . . |
| | . . . |\
| | . . . | 1. the_w : World
| | . . . | 2. FairFifo
| | . . . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . | 5. ∀e:E.
| | . . . |    ¬first(e)
| | . . . |    ⇒
| | . . . |    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . |    & (∀e':E.
| | . . . |       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . | 6. ∀e:E.
| | . . . |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . . | 7. Trans e,e':E. e <c e'
| | . . . | 8. SWellFounded(e <c e')
| | . . . | 9. ∀e:E.
| | . . . |    isrcv(kind(e))
| | . . . |    ⇒
| | . . . |    sends(lnk(kind(e));sender(e))[index(e)]
| | . . . |    =
| | . . . |    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . | 11. ∀e,e':E.
| | . . . |     e <c e'
| | . . . |     ⇒
| | . . . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . . | 12. ∀e:E.
| | . . . |     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . | 14. e : E
| | . . . | 15. e' : E
| | . . . | 16. isrcv(kind(e))
| | . . . | 17. isrcv(kind(e'))
| | . . . | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . | 19. t' : ℕ
| | . . . | 20. mu(λt.match(lnk(kind(e'));t;time(e'))) = t'
```

```
| | . . . | ⊢  ∃n:ℕ. (λt.match(lnk(kind(e));t;time(e)))(n)
| | . . . |   by Reduce 0
| | . . . |      THEN
| | . . . |      Inst
| | . . . |      Thm* ∀the_w:World, e:E.
| | . . . |           FairFifo
| | . . . |           ⇒
| | . . . |           isrcv(kind(e))
| | . . . |           ⇒
| | . . . |           (∃t:ℕtime(e). match(lnk(kind(e));t;time(e)))
| | . . . |      {w_DASH_match_DASH_exists}
| | . . . |      [the_w;e]
| | . . . |      THEN
| | . . . |      ParallelOp -1
| | . . . |
| | . . . .\
| | . . . . 1. the_w : World
| | . . . . 2. FairFifo
| | . . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . 5. ∀e:E.
| | . . . . |  ¬first(e)
| | . . . . |  ⇒
| | . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . |  & (∀e':E.
| | . . . . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . 6. ∀e:E.
| | . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . 7. Trans e,e':E. e <c e'
| | . . . . 8. SWellFounded(e <c e')
| | . . . . 9. ∀e:E.
| | . . . . |  isrcv(kind(e))
| | . . . . |  ⇒
| | . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . |  =
| | . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . 11. ∀e,e':E.
| | . . . . |   e <c e'
| | . . . . |   ⇒
| | . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . . . 12. ∀e:E.
| | . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . 14. e : E
| | . . . . 15. e' : E
| | . . . . 16. isrcv(kind(e))
| | . . . . 17. isrcv(kind(e'))
| | . . . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . 19. t' : ℕ
| | . . . . 20. mu(λt.match(lnk(kind(e'));t;time(e'))) = t'
| | . . . . 21. t : ℕ
| | . . . . 22. mu(λt.match(lnk(kind(e));t;time(e))) = t
| | . . . . ⊢  match(lnk(kind(e));t;time(e))
```

```
| | . . . . |  ⇒
| | . . . . |  match(lnk(kind(e'));t';time(e'))
| | . . . . |  ⇒
| | . . . . |  (e <loc e'
| | . . . . |   ⟺
| | . . . . |   <source(lnk(kind(e))),t> <loc <source(lnk(kind(e')))
| | . . . . |                                  ,t'>
| | . . . . |   ∨ <source(lnk(kind(e))),t> = <source(lnk(kind(e')))
| | . . . . |                                 ,t'>
| | . . . . |     & ||rcvs(lnk(kind(e));time(e))||
| | . . . . |       -||snds(lnk(kind(e));time(...))||<||...||
| | . . . . |       -||snds(lnk(kind(e'));time(<source(lnk(kind(e')))
| | . . . . |                                   ,t'>))||)
| | . . . . | by Thin -1 THEN Thin -2
| | . . . . |
| | . . . . 1. the_w : World
| | . . . . 2. FairFifo
| | . . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . 5. ∀e:E.
| | . . . . |  ¬first(e)
| | . . . . |  ⇒
| | . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . |  & (∀e':E.
| | . . . . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . 6. ∀e:E.
| | . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . 7. Trans e,e':E. e <c e'
| | . . . . 8. SWellFounded(e <c e')
| | . . . . 9. ∀e:E.
| | . . . . |  isrcv(kind(e))
| | . . . . |  ⇒
| | . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . |  =
| | . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . 11. ∀e,e':E.
| | . . . . |   e <c e'
| | . . . . |   ⇒
| | . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . . . 12. ∀e:E.
| | . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . 14. e : E
| | . . . . 15. e' : E
| | . . . . 16. isrcv(kind(e))
| | . . . . 17. isrcv(kind(e'))
| | . . . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . 19. t' : ℕ
| | . . . . 20. t : ℕ
| | . . . . ⊢  match(lnk(kind(e));t;time(e))
| | . . . . |  ⇒
| | . . . . |  match(lnk(kind(e'));t';time(e'))
| | . . . . |  ⇒
```

```
| | . . . . |  (e <loc e'
| | . . . . |   ⟺
| | . . . . |   <source(lnk(kind(e))),t> <loc <source(lnk(kind(e')))
| | . . . . |                                 ,t'>
| | . . . . |   ∨ <source(lnk(kind(e))),t> = <source(lnk(kind(e')))
| | . . . . |                                ,t'>
| | . . . . |     & ||rcvs(lnk(kind(e));time(e))||
| | . . . . |       -||snds(lnk(kind(e));time(...))||<||...||
| | . . . . |       -||snds(lnk(kind(e'));time(<source(lnk(kind(e')))
| | . . . . |                                   ,t'>))||)
| | . . . . | by Unfold `w-locl` 0 THEN Reduce 0
| | . . . . |
| | . . . . 1. the_w : World
| | . . . . 2. FairFifo
| | . . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . 5. ∀e:E.
| | . . . . |  ¬first(e)
| | . . . . |  ⇒
| | . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . |  & (∀e':E.
| | . . . . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . 6. ∀e:E.
| | . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . 7. Trans e,e':E. e <c e'
| | . . . . 8. SWellFounded(e <c e')
| | . . . . 9. ∀e:E.
| | . . . . |  isrcv(kind(e))
| | . . . . |  ⇒
| | . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . |  =
| | . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . 11. ∀e,e':E.
| | . . . . |   e <c e'
| | . . . . |   ⇒
| | . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . . . 12. ∀e:E.
| | . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . 14. e : E
| | . . . . 15. e' : E
| | . . . . 16. isrcv(kind(e))
| | . . . . 17. isrcv(kind(e'))
| | . . . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . 19. t' : ℕ
| | . . . . 20. t : ℕ
| | . . . . ⊢  match(lnk(kind(e));t;time(e))
| | . . . . |  ⇒
| | . . . . |  match(lnk(kind(e'));t';time(e'))
| | . . . . |  ⇒
| | . . . . |  (loc(e) = loc(e') & time(e)<time(e')
| | . . . . |   ⟺
| | . . . . |   source(lnk(kind(e))) = source(lnk(kind(e'))) & t<t'
```

```
| | . . . . |   ∨ <source(lnk(kind(e))),t> = <source(lnk(kind(e')))
| | . . . . |                                 ,t'>
| | . . . . |     & ||rcvs(lnk(kind(e));time(e))||
| | . . . . |       -||snds(lnk(kind(e));t)||<||rcvs(lnk(...);...)||
| | . . . . |       -||snds(lnk(kind(e'));t')||)
| | . . . . | by Unfold `w-eq-E` 0 THEN Reduce 0
| | . . . . |    THEN
| | . . . . |    RW assert_pushdownC 0
| | . . . . |
| | . . . . 1. the_w : World
| | . . . . 2. FairFifo
| | . . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . 5. ∀e:E.
| | . . . . |  ¬first(e)
| | . . . . |  ⇒
| | . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . |  & (∀e':E.
| | . . . . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . 6. ∀e:E.
| | . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . 7. Trans e,e':E. e <c e'
| | . . . . 8. SWellFounded(e <c e')
| | . . . . 9. ∀e:E.
| | . . . . |  isrcv(kind(e))
| | . . . . |  ⇒
| | . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . |  =
| | . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . 11. ∀e,e':E.
| | . . . . |   e <c e'
| | . . . . |   ⇒
| | . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . . . 12. ∀e:E.
| | . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . 14. e : E
| | . . . . 15. e' : E
| | . . . . 16. isrcv(kind(e))
| | . . . . 17. isrcv(kind(e'))
| | . . . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . 19. t' : ℕ
| | . . . . 20. t : ℕ
| | . . . . ⊢  match(lnk(kind(e));t;time(e))
| | . . . . |  ⇒
| | . . . . |  match(lnk(kind(e'));t';time(e'))
| | . . . . |  ⇒
| | . . . . |  (loc(e) = loc(e') & time(e)<time(e')
| | . . . . |   ⟺
| | . . . . |   source(lnk(kind(e))) = source(lnk(kind(e'))) & t<t'
| | . . . . |   ∨ source(lnk(kind(e))) = source(lnk(kind(e')))
| | . . . . |     & t = t'
| | . . . . |     & ||rcvs(lnk(kind(e));time(e))||
```

```
| | . . . . |       -||snds(lnk(kind(e));t)||<||rcvs(lnk(...);...)||
| | . . . . |       -||snds(lnk(kind(e'));t')||)
| | . . . . | by RW assert_pushdownC 0
| | . . . . |
| | . . . . 1. the_w : World
| | . . . . 2. FairFifo
| | . . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . 5. ∀e:E.
| | . . . . |  ¬first(e)
| | . . . . |  ⇒
| | . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . |  & (∀e':E.
| | . . . . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . 6. ∀e:E.
| | . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . 7. Trans e,e':E. e <c e'
| | . . . . 8. SWellFounded(e <c e')
| | . . . . 9. ∀e:E.
| | . . . . |  isrcv(kind(e))
| | . . . . |  ⇒
| | . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . |  =
| | . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . 11. ∀e,e':E.
| | . . . . |   e <c e'
| | . . . . |   ⇒
| | . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . . . 12. ∀e:E.
| | . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . 14. e : E
| | . . . . 15. e' : E
| | . . . . 16. isrcv(kind(e))
| | . . . . 17. isrcv(kind(e'))
| | . . . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . 19. t' : ℕ
| | . . . . 20. t : ℕ
| | . . . . ⊢  match(lnk(kind(e));t;time(e))
| | . . . . |  ⇒
| | . . . . |  match(lnk(kind(e'));t';time(e'))
| | . . . . |  ⇒
| | . . . . |  (loc(e) = loc(e') & time(e)<time(e')
| | . . . . |   ⟺
| | . . . . |   source(lnk(kind(e))) = source(lnk(kind(e'))) & t<t'
| | . . . . |   ∨ source(lnk(kind(e))) = source(lnk(kind(e')))
| | . . . . |     & t = t'
| | . . . . |     & ||rcvs(lnk(kind(e));time(e))||
| | . . . . |       -||snds(lnk(kind(e));t)||<||rcvs(lnk(...);...)||
| | . . . . |       -||snds(lnk(kind(e'));t')||)
| | . . . . | by HypSubst -3 0 THEN GenConcl (lnk(kind(e')) = l)
| | . . . . |
| | . . . . 1. the_w : World
```

```
| | . . . . . 2. FairFifo
| | . . . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . . 5. ∀e:E.
| | . . . . . |  ¬first(e)
| | . . . . . |  ⇒
| | . . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . . |  & (∀e':E.
| | . . . . . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . . 6. ∀e:E.
| | . . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . . 7. Trans e,e':E. e <c e'
| | . . . . . 8. SWellFounded(e <c e')
| | . . . . . 9. ∀e:E.
| | . . . . . |  isrcv(kind(e))
| | . . . . . |  ⇒
| | . . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . . |  =
| | . . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . . 11. ∀e,e':E.
| | . . . . . |   e <c e'
| | . . . . . |   ⇒
| | . . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . . . . 12. ∀e:E.
| | . . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . . . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . . 14. e : E
| | . . . . . 15. e' : E
| | . . . . . 16. isrcv(kind(e))
| | . . . . . 17. isrcv(kind(e'))
| | . . . . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . . 19. t' : ℕ
| | . . . . . 20. t : ℕ
| | . . . . . 21. l : IdLnk
| | . . . . . 22. lnk(kind(e')) = l
| | . . . . . ⊢  match(l;t;time(e))
| | . . . . . |  ⇒
| | . . . . . |  match(l;t';time(e'))
| | . . . . . |  ⇒
| | . . . . . |  (loc(e) = loc(e') & time(e)<time(e')
| | . . . . . |   ⟺
| | . . . . . |   source(l) = source(l) & t<t'
| | . . . . . |   ∨ source(l) = source(l) & t = t'
| | . . . . . |     & ||rcvs(l;time(e))||
| | . . . . . |       -||snds(l;t)||<||rcvs(l;time(e'))||
| | . . . . . |       -||snds(l;t')||)  by Auto
| | . . . . . |
| | . . . . . |\
| | . . . . . | 1. the_w : World
| | . . . . . | 2. FairFifo
| | . . . . . | 3. ∀e,e':E.
| | . . . . . | |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . . | 4. ∀e:E.
```

```
| | . . . . | |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | 5. ∀e:E.
| | . . . . | |  ¬first(e)
| | . . . . | |  ⇒
| | . . . . | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | |  & (∀e':E.
| | . . . . | |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . | 6. ∀e:E.
| | . . . . | |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | 7. Trans e,e':E. e <c e'
| | . . . . | 8. SWellFounded(e <c e')
| | . . . . | 9. ∀e:E.
| | . . . . | |  isrcv(kind(e))
| | . . . . | |  ⇒
| | . . . . | |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | |  =
| | . . . . | |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | 11. ∀e,e':E.
| | . . . . | |   e <c e'
| | . . . . | |   ⇒
| | . . . . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . . . | 12. ∀e:E.
| | . . . . | |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . . . | 13. ∀e:E, l:IdLnk.
| | . . . . | |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | 14. e : E
| | . . . . | 15. e' : E
| | . . . . | 16. isrcv(kind(e))
| | . . . . | 17. isrcv(kind(e'))
| | . . . . | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | 19. t' : ℕ
| | . . . . | 20. t : ℕ
| | . . . . | 21. l : IdLnk
| | . . . . | 22. lnk(kind(e')) = l
| | . . . . | 23. match(l;t;time(e))
| | . . . . | 24. match(l;t';time(e'))
| | . . . . | 25. loc(e) = loc(e')
| | . . . . | 26. time(e)<time(e')
| | . . . . | ⊢  source(l) = source(l) & t<t'
| | . . . . | |  ∨ source(l) = source(l) & t = t'
| | . . . . | |    & ||rcvs(l;time(e))||
| | . . . . | |      -||snds(l;t)||<||rcvs(l;time(e'))||
| | . . . . | |      -||snds(l;t')||
| | . . . . | | by AllHyps
| | . . . . | |    (λi.
| | . . . . | |     RWO
| | . . . . | |     Thm* ∀the_w:World, l:IdLnk, t,t':ℕ.
| | . . . . | |          match(l;t;t')
| | . . . . | |          ⟺
| | . . . . | |          ||snds(l;t)||≤||rcvs(l;t')||
| | . . . . | |          & ||rcvs(l;t')||<||snds(l;t)||
| | . . . . | |            +||onlnk(l;m(source(l);t))||
| | . . . . | |     {assert_DASH_w_DASH_match}
```

```
| | . . . . | |     i
| | . . . . | |     THEN
| | . . . . | |     ExRepD)
| | . . . . | |
| | . . . . | 1. the_w : World
| | . . . . | 2. FairFifo
| | . . . . | 3. ∀e,e':E.
| | . . . . | |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | 4. ∀e:E.
| | . . . . | |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | 5. ∀e:E.
| | . . . . | |  ¬first(e)
| | . . . . | |  ⇒
| | . . . . | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | |  & (∀e':E.
| | . . . . | |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . | 6. ∀e:E.
| | . . . . | |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | 7. Trans e,e':E. e <c e'
| | . . . . | 8. SWellFounded(e <c e')
| | . . . . | 9. ∀e:E.
| | . . . . | |  isrcv(kind(e))
| | . . . . | |  ⇒
| | . . . . | |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | |  =
| | . . . . | |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | 11. ∀e,e':E.
| | . . . . | |   e <c e'
| | . . . . | |   ⇒
| | . . . . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . . . | 12. ∀e:E.
| | . . . . | |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . . . | 13. ∀e:E, l:IdLnk.
| | . . . . | |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | 14. e : E
| | . . . . | 15. e' : E
| | . . . . | 16. isrcv(kind(e))
| | . . . . | 17. isrcv(kind(e'))
| | . . . . | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | 19. t' : ℕ
| | . . . . | 20. t : ℕ
| | . . . . | 21. l : IdLnk
| | . . . . | 22. lnk(kind(e')) = l
| | . . . . | 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | |   +||onlnk(l;m(source(l);t))||
| | . . . . | 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | |   +||onlnk(l;m(source(l);t'))||
| | . . . . | 27. loc(e) = loc(e')
| | . . . . | 28. time(e)<time(e')
| | . . . . | ⊢  source(l) = source(l) & t<t'
| | . . . . | |  ∨ source(l) = source(l) & t = t'
```

```
| | . . . . | |    & ||rcvs(l;time(e))||
| | . . . . | |      -||snds(l;t)||<||rcvs(l;time(e'))||
| | . . . . | |      -||snds(l;t')||
| | . . . . | | by Assert (||rcvs(l;time(e))||<||rcvs(l;time(e'))||)
| | . . . . | |
| | . . . . | |\
| | . . . . | | .....assertion .....
| | . . . . | | 1. the_w : World
| | . . . . | | 2. FairFifo
| | . . . . | | 3. ∀e,e':E.
| | . . . . | | |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | | 4. ∀e:E.
| | . . . . | | |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | | 5. ∀e:E.
| | . . . . | | |  ¬first(e)
| | . . . . | | |  ⇒
| | . . . . | | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | | |  & (∀e':E.
| | . . . . | | |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . | | 6. ∀e:E.
| | . . . . | | |  ¬first(e)
| | . . . . | | |  ⇒
| | . . . . | | |  (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | | 7. Trans e,e':E. e <c e'
| | . . . . | | 8. SWellFounded(e <c e')
| | . . . . | | 9. ∀e:E.
| | . . . . | | |  isrcv(kind(e))
| | . . . . | | |  ⇒
| | . . . . | | |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | | |  =
| | . . . . | | |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | | 11. ∀e,e':E.
| | . . . . | | |   e <c e'
| | . . . . | | |   ⇒
| | . . . . | | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | | |   ∨ isrcv(kind(e'))
| | . . . . | | |     & e <c sender(e') ∨ e = sender(e')
| | . . . . | | 12. ∀e:E.
| | . . . . | | |   isrcv(kind(e))
| | . . . . | | |   ⇒
| | . . . . | | |   loc(e) = destination(lnk(kind(e)))
| | . . . . | | 13. ∀e:E, l:IdLnk.
| | . . . . | | |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | | 14. e : E
| | . . . . | | 15. e' : E
| | . . . . | | 16. isrcv(kind(e))
| | . . . . | | 17. isrcv(kind(e'))
| | . . . . | | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | | 19. t' : ℕ
| | . . . . | | 20. t : ℕ
| | . . . . | | 21. l : IdLnk
| | . . . . | | 22. lnk(kind(e')) = l
| | . . . . | | 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | | 24. ||rcvs(l;time(e))||<||snds(l;t)||
```

```
| | . . . . | | |   +||onlnk(l;m(source(l);t))||
| | . . . . | | 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | | 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | | |   +||onlnk(l;m(source(l);t'))||
| | . . . . | | 27. loc(e) = loc(e')
| | . . . . | | 28. time(e)<time(e')
| | . . . . | | ⊢  ||rcvs(l;time(e))||<||rcvs(l;time(e'))||
| | . . . . | | | by Unfold `w-rcvs` 0
| | . . . . | | |    THEN
| | . . . . | | |    Assert isrcv(l;a(destination(l);time(e)))
| | . . . . | | |
| | . . . . | | |\
| | . . . . | | | .....assertion .....
| | . . . . | | | 1. the_w : World
| | . . . . | | | 2. FairFifo
| | . . . . | | | 3. ∀e,e':E.
| | . . . . | | | |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | | | 4. ∀e:E.
| | . . . . | | | |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | | | 5. ∀e:E.
| | . . . . | | | |  ¬first(e)
| | . . . . | | | |  ⇒
| | . . . . | | | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | | | |  & (∀e':E.
| | . . . . | | | |     loc(e') = loc(e)
| | . . . . | | | |     ⇒
| | . . . . | | | |     ¬(pred(e) <c e' & e' <c e))
| | . . . . | | | 6. ∀e:E.
| | . . . . | | | |  ¬first(e)
| | . . . . | | | |  ⇒
| | . . . . | | | |  (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | | | 7. Trans e,e':E. e <c e'
| | . . . . | | | 8. SWellFounded(e <c e')
| | . . . . | | | 9. ∀e:E.
| | . . . . | | | |  isrcv(kind(e))
| | . . . . | | | |  ⇒
| | . . . . | | | |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | | | |  =
| | . . . . | | | |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | | | 11. ∀e,e':E.
| | . . . . | | | |   e <c e'
| | . . . . | | | |   ⇒
| | . . . . | | | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | | | |   ∨ isrcv(kind(e'))
| | . . . . | | | |     & e <c sender(e') ∨ e = sender(e')
| | . . . . | | | 12. ∀e:E.
| | . . . . | | | |   isrcv(kind(e))
| | . . . . | | | |   ⇒
| | . . . . | | | |   loc(e) = destination(lnk(kind(e)))
| | . . . . | | | 13. ∀e:E, l:IdLnk.
| | . . . . | | | |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | | | 14. e : E
| | . . . . | | | 15. e' : E
| | . . . . | | | 16. isrcv(kind(e))
```

```
| | . . . . | | | 17. isrcv(kind(e'))
| | . . . . | | | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | | | 19. t' : ℕ
| | . . . . | | | 20. t : ℕ
| | . . . . | | | 21. l : IdLnk
| | . . . . | | | 22. lnk(kind(e')) = l
| | . . . . | | | 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | | | 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | | | |   +||onlnk(l;m(source(l);t))||
| | . . . . | | | 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | | | 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | | | |   +||onlnk(l;m(source(l);t'))||
| | . . . . | | | 27. loc(e) = loc(e')
| | . . . . | | | 28. time(e)<time(e')
| | . . . . | | | ⊢  isrcv(l;a(destination(l);time(e)))
| | . . . . | | | | by Subst (destination(l) = loc(e)) 0
| | . . . . | | | |
| | . . . . | | | |\
| | . . . . | | | | .....equality .....
| | . . . . | | | | 1. the_w : World
| | . . . . | | | | 2. FairFifo
| | . . . . | | | | 3. ∀e,e':E.
| | . . . . | | | | |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | | | | 4. ∀e:E.
| | . . . . | | | | |  first(e)
| | . . . . | | | | |  ⟺
| | . . . . | | | | |  (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | | | | 5. ∀e:E.
| | . . . . | | | | |  ¬first(e)
| | . . . . | | | | |  ⇒
| | . . . . | | | | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | | | | |  & (∀e':E.
| | . . . . | | | | |     loc(e') = loc(e)
| | . . . . | | | | |     ⇒
| | . . . . | | | | |     ¬(pred(e) <c e' & e' <c e))
| | . . . . | | | | 6. ∀e:E.
| | . . . . | | | | |  ¬first(e)
| | . . . . | | | | |  ⇒
| | . . . . | | | | |  (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | | | | 7. Trans e,e':E. e <c e'
| | . . . . | | | | 8. SWellFounded(e <c e')
| | . . . . | | | | 9. ∀e:E.
| | . . . . | | | | |  isrcv(kind(e))
| | . . . . | | | | |  ⇒
| | . . . . | | | | |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | | | | |  =
| | . . . . | | | | |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | | | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | | | | 11. ∀e,e':E.
| | . . . . | | | | |   e <c e'
| | . . . . | | | | |   ⇒
| | . . . . | | | | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | | | | |   ∨ isrcv(kind(e'))
| | . . . . | | | | |     & e <c sender(e') ∨ e = sender(e')
| | . . . . | | | | 12. ∀e:E.
```

```
| | . . . . | | | | |   isrcv(kind(e))
| | . . . . | | | | |   ⇒
| | . . . . | | | | |   loc(e) = destination(lnk(kind(e)))
| | . . . . | | | | | 13. ∀e:E, l:IdLnk.
| | . . . . | | | | |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | | | | | 14. e : E
| | . . . . | | | | | 15. e' : E
| | . . . . | | | | | 16. isrcv(kind(e))
| | . . . . | | | | | 17. isrcv(kind(e'))
| | . . . . | | | | | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | | | | | 19. t' : ℕ
| | . . . . | | | | | 20. t : ℕ
| | . . . . | | | | | 21. l : IdLnk
| | . . . . | | | | | 22. lnk(kind(e')) = l
| | . . . . | | | | | 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | | | | | 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | | | | |   +||onlnk(l;m(source(l);t))||
| | . . . . | | | | | 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | | | | | 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | | | | |   +||onlnk(l;m(source(l);t'))||
| | . . . . | | | | | 27. loc(e) = loc(e')
| | . . . . | | | | | 28. time(e)<time(e')
| | . . . . | | | | | ⊢ destination(l) = loc(e)
| | . . . . | | | | | by InstHyp [e] 12
| | . . . . | | | | |    THEN
| | . . . . | | | | |    Assert
| | . . . . | | | | |    (destination(l) = destination(lnk(kind(e))))
| | . . . . | | | | |
| | . . . . | | | | .....assertion .....
| | . . . . | | | | 1. the_w : World
| | . . . . | | | | 2. FairFifo
| | . . . . | | | | 3. ∀e,e':E.
| | . . . . | | | |   loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | | | | 4. ∀e:E.
| | . . . . | | | |   first(e)
| | . . . . | | | |   ⟺
| | . . . . | | | |   (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | | | | 5. ∀e:E.
| | . . . . | | | |   ¬first(e)
| | . . . . | | | |   ⇒
| | . . . . | | | |   loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | | | |   & (∀e':E.
| | . . . . | | | |      loc(e') = loc(e)
| | . . . . | | | |      ⇒
| | . . . . | | | |      ¬(pred(e) <c e' & e' <c e))
| | . . . . | | | | 6. ∀e:E.
| | . . . . | | | |   ¬first(e)
| | . . . . | | | |   ⇒
| | . . . . | | | |   (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | | | | 7. Trans e,e':E. e <c e'
| | . . . . | | | | 8. SWellFounded(e <c e')
| | . . . . | | | | 9. ∀e:E.
| | . . . . | | | |   isrcv(kind(e))
| | . . . . | | | |   ⇒
| | . . . . | | | |   sends(lnk(kind(e));sender(e))[index(e)]
```

```
| | . . . . | | | |     =
| | . . . . | | | |     msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | | | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | | | | 11. ∀e,e':E.
| | . . . . | | | |     e <c e'
| | . . . . | | | |     ⇒
| | . . . . | | | |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | | | |     ∨ isrcv(kind(e'))
| | . . . . | | | |       & e <c sender(e') ∨ e = sender(e')
| | . . . . | | | | 12. ∀e:E.
| | . . . . | | | |     isrcv(kind(e))
| | . . . . | | | |     ⇒
| | . . . . | | | |     loc(e) = destination(lnk(kind(e)))
| | . . . . | | | | 13. ∀e:E, l:IdLnk.
| | . . . . | | | |     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | | | | 14. e : E
| | . . . . | | | | 15. e' : E
| | . . . . | | | | 16. isrcv(kind(e))
| | . . . . | | | | 17. isrcv(kind(e'))
| | . . . . | | | | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | | | | 19. t' : ℕ
| | . . . . | | | | 20. t : ℕ
| | . . . . | | | | 21. l : IdLnk
| | . . . . | | | | 22. lnk(kind(e')) = l
| | . . . . | | | | 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | | | | 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | | | |     +||onlnk(l;m(source(l);t))||
| | . . . . | | | | 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | | | | 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | | | |     +||onlnk(l;m(source(l);t'))||
| | . . . . | | | | 27. loc(e) = loc(e')
| | . . . . | | | | 28. time(e)<time(e')
| | . . . . | | | | 29. loc(e) = destination(lnk(kind(e)))
| | . . . . | | | | ⊢  destination(l) = destination(lnk(kind(e)))
| | . . . . | | | |   by Analyze
| | . . . . | | | |
| | . . . . | | | .\
| | . . . . | | | . 1. the_w : World
| | . . . . | | | . 2. FairFifo
| | . . . . | | | . 3. ∀e,e':E.
| | . . . . | | | . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | | | . 4. ∀e:E.
| | . . . . | | | . |  first(e)
| | . . . . | | | . |  ⟺
| | . . . . | | | . |  (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | | | . 5. ∀e:E.
| | . . . . | | | . |  ¬first(e)
| | . . . . | | | . |  ⇒
| | . . . . | | | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | | | . |  & (∀e':E.
| | . . . . | | | . |     loc(e') = loc(e)
| | . . . . | | | . |     ⇒
| | . . . . | | | . |     ¬(pred(e) <c e' & e' <c e))
| | . . . . | | | . 6. ∀e:E.
| | . . . . | | | . |  ¬first(e)
```

```
| | . . . . | | | . |  ⇒
| | . . . . | | | . |  (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | | | . 7. Trans e,e':E. e <c e'
| | . . . . | | | . 8. SWellFounded(e <c e')
| | . . . . | | | . 9. ∀e:E.
| | . . . . | | | . |  isrcv(kind(e))
| | . . . . | | | . |  ⇒
| | . . . . | | | . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | | | . |  =
| | . . . . | | | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | | | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | | | . 11. ∀e,e':E.
| | . . . . | | | . |   e <c e'
| | . . . . | | | . |   ⇒
| | . . . . | | | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | | | . |   ∨ isrcv(kind(e'))
| | . . . . | | | . |     & e <c sender(e') ∨ e = sender(e')
| | . . . . | | | . 12. ∀e:E.
| | . . . . | | | . |   isrcv(kind(e))
| | . . . . | | | . |   ⇒
| | . . . . | | | . |   loc(e) = destination(lnk(kind(e)))
| | . . . . | | | . 13. ∀e:E, l:IdLnk.
| | . . . . | | | . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | | | . 14. e : E
| | . . . . | | | . 15. e' : E
| | . . . . | | | . 16. isrcv(kind(e))
| | . . . . | | | . 17. isrcv(kind(e'))
| | . . . . | | | . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | | | . 19. t' : ℕ
| | . . . . | | | . 20. t : ℕ
| | . . . . | | | . 21. l : IdLnk
| | . . . . | | | . 22. lnk(kind(e')) = l
| | . . . . | | | . 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | | | . 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | | | . |   +||onlnk(l;m(source(l);t))||
| | . . . . | | | . 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | | | . 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | | | . |   +||onlnk(l;m(source(l);t'))||
| | . . . . | | | . 27. loc(e) = loc(e')
| | . . . . | | | . 28. time(e)<time(e')
| | . . . . | | | . ⊢  isrcv(l;a(loc(e);time(e)))
| | . . . . | | | . | by Unfold `w-isrcvl` 0
| | . . . . | | | . |    THEN
| | . . . . | | | . |    AutoBoolCase isnull(a(loc(e);time(e)))
| | . . . . | | | . |
| | . . . . | | | . |\
| | . . . . | | | . | 1. the_w : World
| | . . . . | | | . | 2. FairFifo
| | . . . . | | | . | 3. ∀e,e':E.
| | . . . . | | | . |    loc(e) = loc(e')
| | . . . . | | | . |    ⇒
| | . . . . | | | . |    e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | | | . | 4. ∀e:E.
| | . . . . | | | . |    first(e)
| | . . . . | | | . |    ⟺
```

```
| | . . . . | | | . |    (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | | | . | 5. ∀e:E.
| | . . . . | | | . |    ¬first(e)
| | . . . . | | | . |    ⇒
| | . . . . | | | . |    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | | | . |    & (∀e':E.
| | . . . . | | | . |       loc(e') = loc(e)
| | . . . . | | | . |       ⇒
| | . . . . | | | . |       ¬(pred(e) <c e' & e' <c e))
| | . . . . | | | . | 6. ∀e:E.
| | . . . . | | | . |    ¬first(e)
| | . . . . | | | . |    ⇒
| | . . . . | | | . |    (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | | | . | 7. Trans e,e':E. e <c e'
| | . . . . | | | . | 8. SWellFounded(e <c e')
| | . . . . | | | . | 9. ∀e:E.
| | . . . . | | | . |    isrcv(kind(e))
| | . . . . | | | . |    ⇒
| | . . . . | | | . |    sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | | | . |    =
| | . . . . | | | . |    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | | | . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | | | . | 11. ∀e,e':E.
| | . . . . | | | . |     e <c e'
| | . . . . | | | . |     ⇒
| | . . . . | | | . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | | | . |     ∨ isrcv(kind(e'))
| | . . . . | | | . |       & e <c sender(e') ∨ e = sender(e')
| | . . . . | | | . | 12. ∀e:E.
| | . . . . | | | . |     isrcv(kind(e))
| | . . . . | | | . |     ⇒
| | . . . . | | | . |     loc(e) = destination(lnk(kind(e)))
| | . . . . | | | . | 13. ∀e:E, l:IdLnk.
| | . . . . | | | . |     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | | | . | 14. e : E
| | . . . . | | | . | 15. e' : E
| | . . . . | | | . | 16. isrcv(kind(e))
| | . . . . | | | . | 17. isrcv(kind(e'))
| | . . . . | | | . | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | | | . | 19. t' : ℕ
| | . . . . | | | . | 20. t : ℕ
| | . . . . | | | . | 21. l : IdLnk
| | . . . . | | | . | 22. lnk(kind(e')) = l
| | . . . . | | | . | 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | | | . | 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | | | . |     +||onlnk(l;m(source(l);t))||
| | . . . . | | | . | 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | | | . | 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | | | . |     +||onlnk(l;m(source(l);t'))||
| | . . . . | | | . | 27. loc(e) = loc(e')
| | . . . . | | | . | 28. time(e)<time(e')
| | . . . . | | | . | 29. isnull(a(loc(e);time(e)))
| | . . . . | | | . | ⊢  False
| | . . . . | | | . |   by DVar `e' THEN DVar `e' THEN All Reduce
| | . . . . | | | . |
```

```
| | . . . . | | | . .\
| | . . . . | | | . . ...
| | . . . . | | | . . |
| | . . . . | | | . . |\
| | . . . . | | | . . | 1. the_w : World
| | . . . . | | | . . | 2. FairFifo
| | . . . . | | | . . | 3. ∀e,e':E.
| | . . . . | | | . . |    loc(e) = loc(e')
| | . . . . | | | . . |    ⇒
| | . . . . | | | . . |    e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | | | . . | 4. ∀e:E.
| | . . . . | | | . . |    first(e)
| | . . . . | | | . . |    ⟺
| | . . . . | | | . . |    (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | | | . . | 5. ∀e:E.
| | . . . . | | | . . |    ¬first(e)
| | . . . . | | | . . |    ⇒
| | . . . . | | | . . |    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | | | . . |    & (∀e':E.
| | . . . . | | | . . |       loc(e') = loc(e)
| | . . . . | | | . . |       ⇒
| | . . . . | | | . . |       ¬(pred(e) <c e' & e' <c e))
| | . . . . | | | . . | 6. ∀e:E.
| | . . . . | | | . . |    ¬first(e)
| | . . . . | | | . . |    ⇒
| | . . . . | | | . . |    (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | | | . . | 7. Trans e,e':E. e <c e'
| | . . . . | | | . . | 8. SWellFounded(e <c e')
| | . . . . | | | . . | 9. ∀e:E.
| | . . . . | | | . . |    isrcv(kind(e))
| | . . . . | | | . . |    ⇒
| | . . . . | | | . . |    sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | | | . . |    =
| | . . . . | | | . . |    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | | | . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | | | . . | 11. ∀e,e':E.
| | . . . . | | | . . |     e <c e'
| | . . . . | | | . . |     ⇒
| | . . . . | | | . . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | | | . . |     ∨ isrcv(kind(e'))
| | . . . . | | | . . |       & e <c sender(e') ∨ e = sender(e')
| | . . . . | | | . . | 12. ∀e:E.
| | . . . . | | | . . |     isrcv(kind(e))
| | . . . . | | | . . |     ⇒
| | . . . . | | | . . |     loc(e) = destination(lnk(kind(e)))
| | . . . . | | | . . | 13. ∀e:E, l:IdLnk.
| | . . . . | | | . . |     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | | | . . | 14. e : E
| | . . . . | | | . . | 15. e' : E
| | . . . . | | | . . | 16. isrcv(kind(e))
| | . . . . | | | . . | 17. isrcv(kind(e'))
| | . . . . | | | . . | 18. lnk(kind(e)) = lnk(kind(e'))

| | . . . . | | | . . | 19. t' : ℕ
| | . . . . | | | . . | 20. t : ℕ
| | . . . . | | | . . | 21. l : IdLnk
```

```
| | . . . . | | | . . | 22. lnk(kind(e')) = l
| | . . . . | | | . . | 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | | | . . | 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | | | . . |     +||onlnk(l;m(source(l);t))||
| | . . . . | | | . . | 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | | | . . | 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | | | . . |     +||onlnk(l;m(source(l);t'))||
| | . . . . | | | . . | 27. loc(e) = loc(e')
| | . . . . | | | . . | 28. time(e)<time(e')
| | . . . . | | | . . | 29. ¬isnull(a(loc(e);time(e)))
| | . . . . | | | . . | 30. isrcv(kind(a(loc(e);time(e))))
| | . . . . | | | . . | ⊢  lnk(kind(a(loc(e);time(e)))) = l
| | . . . . | | | . . |   by RW assert_pushdownC 0
| | . . . . | | | . . |      THEN
| | . . . . | | | . . |      Unfolds [`w-loc`;`w-time`] 0
| | . . . . | | | . . |      THEN
| | . . . . | | | . . |      All
| | . . . . | | | . . |      (λi.
| | . . . . | | | . . |       Unfold `w-ekind` i
| | . . . . | | | . . |       THEN
| | . . . . | | | . . |       Unfold `w-act` i)
| | . . . . | | | . . |
| | . . . . | | | . . .\
| | . . . . | | | . . . 1. the_w : World
| | . . . . | | | . . . 2. FairFifo
| | . . . . | | | . . . 3. ∀e,e':E.
| | . . . . | | | . . .    loc(e) = loc(e')
| | . . . . | | | . . .    ⇒
| | . . . . | | | . . .    e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | | | . . . 4. ∀e:E.
| | . . . . | | | . . .    first(e)
| | . . . . | | | . . .    ⟺
| | . . . . | | | . . .    (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | | | . . . 5. ∀e:E.
| | . . . . | | | . . .    ¬first(e)
| | . . . . | | | . . .    ⇒
| | . . . . | | | . . .    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | | | . . .    & (∀e':E.
| | . . . . | | | . . .       loc(e') = loc(e)
| | . . . . | | | . . .       ⇒
| | . . . . | | | . . .       ¬(pred(e) <c e' & e' <c e))
| | . . . . | | | . . . 6. ∀e:E.
| | . . . . | | | . . .    ¬first(e)
| | . . . . | | | . . .    ⇒
| | . . . . | | | . . .    (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | | | . . . 7. Trans e,e':E. e <c e'
| | . . . . | | | . . . 8. SWellFounded(e <c e')
| | . . . . | | | . . . 9. ∀e:E.
| | . . . . | | | . . .    isrcv(kind(e))
| | . . . . | | | . . .    ⇒
| | . . . . | | | . . .    sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | | | . . .    =
| | . . . . | | | . . .    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | | | . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | | | . . . 11. ∀e,e':E.
```

```
| | . . . . | | | . . .     e <c e'
| | . . . . | | | . . .     ⇒
| | . . . . | | | . . .     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | | | . . .     ∨ isrcv(kind(e'))
| | . . . . | | | . . .       & e <c sender(e') ∨ e = sender(e')
| | . . . . | | | . . . 12. ∀e:E.
| | . . . . | | | . . .     isrcv(kind(e))
| | . . . . | | | . . .     ⇒
| | . . . . | | | . . .     loc(e) = destination(lnk(kind(e)))
| | . . . . | | | . . . 13. ∀e:E, l:IdLnk.
| | . . . . | | | . . .     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | | | . . . 14. e : E
| | . . . . | | | . . . 15. e' : E
| | . . . . | | | . . . 16. isrcv(kind(e))
| | . . . . | | | . . . 17. isrcv(kind(e'))
| | . . . . | | | . . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | | | . . . 19. t' : ℕ
| | . . . . | | | . . . 20. t : ℕ
| | . . . . | | | . . . 21. l : IdLnk
| | . . . . | | | . . . 22. lnk(kind(e')) = l
| | . . . . | | | . . . 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | | | . . . 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | | | . . .     +||onlnk(l;m(source(l);t))||
| | . . . . | | | . . . 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | | | . . . 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | | | . . .     +||onlnk(l;m(source(l);t'))||
| | . . . . | | | . . . 27. loc(e) = loc(e')
| | . . . . | | | . . . 28. time(e)<time(e')
| | . . . . | | | . . . 29. ¬isnull(a(loc(e);time(e)))
| | . . . . | | | . . . 30. ¬isrcv(kind(a(loc(e);time(e))))
| | . . . . | | | . . . ⊢  False
| | . . . . | | | . . .   by Analyze -1 THEN NthHypSq 16
| | . . . . | | | . . .      THEN
| | . . . . | | | . . .      Unfolds [`w-loc`;`w-time`;`w-ekind`] 0
| | . . . . | | | . . .      THEN
| | . . . . | | | . . .      Unfold `w-act` 0
| | . . . . | | | . . .      THEN
| | . . . . | | | . . .      Trivial
| | . . . . | | |
| | . . . . | | .\
| | . . . . | | . 1. the_w : World
| | . . . . | | . 2. FairFifo
| | . . . . | | . 3. ∀e,e':E.
| | . . . . | | . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | | . 4. ∀e:E.
| | . . . . | | . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | | . 5. ∀e:E.
| | . . . . | | . |  ¬first(e)
| | . . . . | | . |  ⇒
| | . . . . | | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | | . |  & (∀e':E.
| | . . . . | | . |     loc(e') = loc(e)
| | . . . . | | . |     ⇒
| | . . . . | | . |     ¬(pred(e) <c e' & e' <c e))
| | . . . . | | . 6. ∀e:E.
```

```
| | . . . . | | . |  ¬first(e)
| | . . . . | | . |  ⇒
| | . . . . | | . |  (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | | . 7. Trans e,e':E. e <c e'
| | . . . . | | . 8. SWellFounded(e <c e')
| | . . . . | | . 9. ∀e:E.
| | . . . . | | . |  isrcv(kind(e))
| | . . . . | | . |  ⇒
| | . . . . | | . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | | . |  =
| | . . . . | | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | | . 11. ∀e,e':E.
| | . . . . | | . |   e <c e'
| | . . . . | | . |   ⇒
| | . . . . | | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | | . |   ∨ isrcv(kind(e'))
| | . . . . | | . |     & e <c sender(e') ∨ e = sender(e')
| | . . . . | | . 12. ∀e:E.
| | . . . . | | . |   isrcv(kind(e))
| | . . . . | | . |   ⇒
| | . . . . | | . |   loc(e) = destination(lnk(kind(e)))
| | . . . . | | . 13. ∀e:E, l:IdLnk.
| | . . . . | | . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | | . 14. e : E
| | . . . . | | . 15. e' : E
| | . . . . | | . 16. isrcv(kind(e))
| | . . . . | | . 17. isrcv(kind(e'))
| | . . . . | | . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | | . 19. t' : ℕ
| | . . . . | | . 20. t : ℕ
| | . . . . | | . 21. l : IdLnk
| | . . . . | | . 22. lnk(kind(e')) = l
| | . . . . | | . 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | | . 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | | . |   +||onlnk(l;m(source(l);t))||
| | . . . . | | . 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | | . 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | | . |   +||onlnk(l;m(source(l);t'))||
| | . . . . | | . 27. loc(e) = loc(e')
| | . . . . | | . 28. time(e)<time(e')
| | . . . . | | . 29. isrcv(l;a(destination(l);time(e)))
| | . . . . | | . ⊢  ||filter(λa.isrcv(l;a);map(λt1.
| | . . . . | | . |  a(destination(l);t1);upto(time(e))))||<||...||
| | . . . . | | . | by GenConcl ((λt1.a(destination(l);t1)) = f)
| | . . . . | | . |    THEN
| | . . . . | | . |    GenConcl ((λa.isrcv(l;a)) = P)
| | . . . . | | . |    THEN
| | . . . . | | . |    Assert (P(f(time(e))))
| | . . . . | | . |
| | . . . . | | . |\
| | . . . . | | . | .....assertion .....
| | . . . . | | . | 1. the_w : World
| | . . . . | | . | 2. FairFifo
| | . . . . | | . | 3. ∀e,e':E.
```

```
| | . . . . | | . |    loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
| | . . . . | | . | 4. ∀e:E.
| | . . . . | | . |    first(e)
| | . . . . | | . |    ⟺
| | . . . . | | . |    (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
| | . . . . | | . | 5. ∀e:E.
| | . . . . | | . |    ¬first(e)
| | . . . . | | . |    ⇒
| | . . . . | | . |    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | | . |    & (∀e’:E.
| | . . . . | | . |       loc(e’) = loc(e)
| | . . . . | | . |       ⇒
| | . . . . | | . |       ¬(pred(e) <c e’ & e’ <c e))
| | . . . . | | . | 6. ∀e:E.
| | . . . . | | . |    ¬first(e)
| | . . . . | | . |    ⇒
| | . . . . | | . |    (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | | . | 7. Trans e,e’:E. e <c e’
| | . . . . | | . | 8. SWellFounded(e <c e’)
| | . . . . | | . | 9. ∀e:E.
| | . . . . | | . |    isrcv(kind(e))
| | . . . . | | . |    ⇒
| | . . . . | | . |    sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | | . |    =
| | . . . . | | . |    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | | . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | | . | 11. ∀e,e’:E.
| | . . . . | | . |     e <c e’
| | . . . . | | . |     ⇒
| | . . . . | | . |     ¬first(e’) & e <c pred(e’) ∨ e = pred(e’)
| | . . . . | | . |     ∨ isrcv(kind(e’))
| | . . . . | | . |       & e <c sender(e’) ∨ e = sender(e’)
| | . . . . | | . | 12. ∀e:E.
| | . . . . | | . |     isrcv(kind(e))
| | . . . . | | . |     ⇒
| | . . . . | | . |     loc(e) = destination(lnk(kind(e)))
| | . . . . | | . | 13. ∀e:E, l:IdLnk.
| | . . . . | | . |     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | | . | 14. e : E
| | . . . . | | . | 15. e’ : E
| | . . . . | | . | 16. isrcv(kind(e))
| | . . . . | | . | 17. isrcv(kind(e’))
| | . . . . | | . | 18. lnk(kind(e)) = lnk(kind(e’))
| | . . . . | | . | 19. t’ : ℕ
| | . . . . | | . | 20. t : ℕ
| | . . . . | | . | 21. l : IdLnk
| | . . . . | | . | 22. lnk(kind(e’)) = l
| | . . . . | | . | 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | | . | 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | | . |     +||onlnk(l;m(source(l);t))||
| | . . . . | | . | 25. ||snds(l;t’)||≤||rcvs(l;time(e’))||
| | . . . . | | . | 26. ||rcvs(l;time(e’))||<||snds(l;t’)||
| | . . . . | | . |     +||onlnk(l;m(source(l);t’))||
| | . . . . | | . | 27. loc(e) = loc(e’)
| | . . . . | | . | 28. time(e)<time(e’)
```

```
| | . . . . | | . | 29. isrcv(l;a(destination(l);time(e)))
| | . . . . | | . | 30. f : ℕ→Action(destination(l))
| | . . . . | | . | 31. (λt1.a(destination(l);t1)) = f
| | . . . . | | . | 32. P : Action(destination(l))→𝔹
| | . . . . | | . | 33. (λa.isrcv(l;a)) = P
| | . . . . | | . | ⊢  P(f(time(e)))
| | . . . . | | . |   by ((RevHypSubst -3 0) THEN (RevHypSubst -1 0))
| | . . . . | | . |      THEN
| | . . . . | | . |      Reduce 0
| | . . . . | | . |
| | . . . . | | . .\
| | . . . . | | . . 1. the_w : World
| | . . . . | | . . 2. FairFifo
| | . . . . | | . . 3. ∀e,e':E.
| | . . . . | | . .    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | | . . 4. ∀e:E.
| | . . . . | | . .    first(e)
| | . . . . | | . .    ⟺
| | . . . . | | . .    (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | | . . 5. ∀e:E.
| | . . . . | | . .    ¬first(e)
| | . . . . | | . .    ⇒
| | . . . . | | . .    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | | . .    & (∀e':E.
| | . . . . | | . .       loc(e') = loc(e)
| | . . . . | | . .       ⇒
| | . . . . | | . .       ¬(pred(e) <c e' & e' <c e))
| | . . . . | | . . 6. ∀e:E.
| | . . . . | | . .    ¬first(e)
| | . . . . | | . .    ⇒
| | . . . . | | . .    (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | | . . 7. Trans e,e':E. e <c e'
| | . . . . | | . . 8. SWellFounded(e <c e')
| | . . . . | | . . 9. ∀e:E.
| | . . . . | | . .    isrcv(kind(e))
| | . . . . | | . .    ⇒
| | . . . . | | . .    sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | | . .    =
| | . . . . | | . .    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | | . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | | . . 11. ∀e,e':E.
| | . . . . | | . .     e <c e'
| | . . . . | | . .     ⇒
| | . . . . | | . .     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | | . .     ∨ isrcv(kind(e'))
| | . . . . | | . .       & e <c sender(e') ∨ e = sender(e')
| | . . . . | | . . 12. ∀e:E.
| | . . . . | | . .     isrcv(kind(e))
| | . . . . | | . .     ⇒
| | . . . . | | . .     loc(e) = destination(lnk(kind(e)))
| | . . . . | | . . 13. ∀e:E, l:IdLnk.
| | . . . . | | . .     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | | . . 14. e : E
| | . . . . | | . . 15. e' : E
| | . . . . | | . . 16. isrcv(kind(e))
```

```
| | . . . . | | . . 17. isrcv(kind(e'))
| | . . . . | | . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | | . . 19. t' : ℕ
| | . . . . | | . . 20. t : ℕ
| | . . . . | | . . 21. l : IdLnk
| | . . . . | | . . 22. lnk(kind(e')) = l
| | . . . . | | . . 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | | . . 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | | . .     +||onlnk(l;m(source(l);t))||
| | . . . . | | . . 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | | . . 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | | . .     +||onlnk(l;m(source(l);t'))||
| | . . . . | | . . 27. loc(e) = loc(e')
| | . . . . | | . . 28. time(e)<time(e')
| | . . . . | | . . 29. isrcv(l;a(destination(l);time(e)))
| | . . . . | | . . 30. f : ℕ→Action(destination(l))
| | . . . . | | . . 31. (λt1.a(destination(l);t1)) = f
| | . . . . | | . . 32. P : Action(destination(l))→𝔹
| | . . . . | | . . 33. (λa.isrcv(l;a)) = P
| | . . . . | | . . 34. P(f(time(e)))
| | . . . . | | . . ⊢  ||filter(P;map(f;upto(time(e))))||<||...||
| | . . . . | | . .   by BackThru
| | . . . . | | . .      Thm* ∀i,j:ℕ.
| | . . . . | | . .           i<j
| | . . . . | | . .           ⇒
| | . . . . | | . .           (∀f:(ℕ→T), P:(T→𝔹).
| | . . . . | | . .            P(f(i))
| | . . . . | | . .            ⇒
| | . . . . | | . .            ||filter(P;map(f;upto(i)))||<||...||)
| | . . . . | | . .      {filter_map_upto}
| | . . . . | |
| | . . . . | .\
| | . . . . | . 1. the_w : World
| | . . . . | . 2. FairFifo
| | . . . . | . 3. ∀e,e':E.
| | . . . . | . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | . 4. ∀e:E.
| | . . . . | . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | . 5. ∀e:E.
| | . . . . | . |  ¬first(e)
| | . . . . | . |  ⇒
| | . . . . | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | . |  & (∀e':E.
| | . . . . | . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . | . 6. ∀e:E.
| | . . . . | . |  ¬first(e)
| | . . . . | . |  ⇒
| | . . . . | . |  (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | . 7. Trans e,e':E. e <c e'
| | . . . . | . 8. SWellFounded(e <c e')
| | . . . . | . 9. ∀e:E.
| | . . . . | . |  isrcv(kind(e))
| | . . . . | . |  ⇒
| | . . . . | . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | . |  =
```

```
| | . . . . | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | . 11. ∀e,e':E.
| | . . . . | . |   e <c e'
| | . . . . | . |   ⇒
| | . . . . | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | . |   ∨ isrcv(kind(e'))
| | . . . . | . |     & e <c sender(e') ∨ e = sender(e')
| | . . . . | . 12. ∀e:E.
| | . . . . | . |   isrcv(kind(e))
| | . . . . | . |   ⇒
| | . . . . | . |   loc(e) = destination(lnk(kind(e)))
| | . . . . | . 13. ∀e:E, l:IdLnk.
| | . . . . | . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | . 14. e : E
| | . . . . | . 15. e' : E
| | . . . . | . 16. isrcv(kind(e))
| | . . . . | . 17. isrcv(kind(e'))
| | . . . . | . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | . 19. t' : ℕ
| | . . . . | . 20. t : ℕ
| | . . . . | . 21. l : IdLnk
| | . . . . | . 22. lnk(kind(e')) = l
| | . . . . | . 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | . 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | . |   +||onlnk(l;m(source(l);t))||
| | . . . . | . 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | . 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | . |   +||onlnk(l;m(source(l);t'))||
| | . . . . | . 27. loc(e) = loc(e')
| | . . . . | . 28. time(e)<time(e')
| | . . . . | . 29. ||rcvs(l;time(e))||<||rcvs(l;time(e'))||
| | . . . . | . ⊢  source(l) = source(l) & t<t'
| | . . . . | . |  ∨ source(l) = source(l) & t = t'
| | . . . . | . |    & ||rcvs(l;time(e))||
| | . . . . | . |      -||snds(l;t)||<||rcvs(l;time(e'))||
| | . . . . | . |      -||snds(l;t')||  by Decide (t<t')
| | . . . . | . |
| | . . . . | . |\
| | . . . . | . | 1. the_w : World
| | . . . . | . | 2. FairFifo
| | . . . . | . | 3. ∀e,e':E.
| | . . . . | . |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | . | 4. ∀e:E.
| | . . . . | . |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | . | 5. ∀e:E.
| | . . . . | . |    ¬first(e)
| | . . . . | . |    ⇒
| | . . . . | . |    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | . |    & (∀e':E.
| | . . . . | . |       loc(e') = loc(e)
| | . . . . | . |       ⇒
| | . . . . | . |       ¬(pred(e) <c e' & e' <c e))
| | . . . . | . | 6. ∀e:E.
| | . . . . | . |    ¬first(e)
```

```
| | . . . . . | . |    ⇒
| | . . . . . | . |    (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . . | . | 7. Trans e,e':E. e <c e'
| | . . . . . | . | 8. SWellFounded(e <c e')
| | . . . . . | . | 9. ∀e:E.
| | . . . . . | . |    isrcv(kind(e))
| | . . . . . | . |    ⇒
| | . . . . . | . |    sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . . | . |    =
| | . . . . . | . |    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . . | . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . . | . | 11. ∀e,e':E.
| | . . . . . | . |     e <c e'
| | . . . . . | . |     ⇒
| | . . . . . | . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . . | . |     ∨ isrcv(kind(e'))
| | . . . . . | . |       & e <c sender(e') ∨ e = sender(e')
| | . . . . . | . | 12. ∀e:E.
| | . . . . . | . |     isrcv(kind(e))
| | . . . . . | . |     ⇒
| | . . . . . | . |     loc(e) = destination(lnk(kind(e)))
| | . . . . . | . | 13. ∀e:E, l:IdLnk.
| | . . . . . | . |     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . . | . | 14. e : E
| | . . . . . | . | 15. e' : E
| | . . . . . | . | 16. isrcv(kind(e))
| | . . . . . | . | 17. isrcv(kind(e'))
| | . . . . . | . | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . . | . | 19. t' : ℕ
| | . . . . . | . | 20. t : ℕ
| | . . . . . | . | 21. l : IdLnk
| | . . . . . | . | 22. lnk(kind(e')) = l
| | . . . . . | . | 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . . | . | 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . . | . |     +||onlnk(l;m(source(l);t))||
| | . . . . . | . | 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . . | . | 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . . | . |     +||onlnk(l;m(source(l);t'))||
| | . . . . . | . | 27. loc(e) = loc(e')
| | . . . . . | . | 28. time(e)<time(e')
| | . . . . . | . | 29. ||rcvs(l;time(e))||<||rcvs(l;time(e'))||
| | . . . . . | . | 30. t<t'
| | . . . . . | . | ⊢  source(l) = source(l) & t<t'
| | . . . . . | . |    ∨ source(l) = source(l) & t = t'
| | . . . . . | . |      & ||rcvs(l;time(e))||
| | . . . . . | . |        -||snds(l;t)||<||rcvs(l;time(e'))||
| | . . . . . | . |        -||snds(l;t')||  by OrLeft
| | . . . . . | . |
| | . . . . . | . .\
| | . . . . . | . . 1. the_w : World
| | . . . . . | . . 2. FairFifo
| | . . . . . | . . 3. ∀e,e':E.
| | . . . . . | . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . . | . . 4. ∀e:E.
| | . . . . . | . . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
```

```
| | . . . . . | . . 5. ∀e:E.
| | . . . . . | . . |  ¬first(e)
| | . . . . . | . . |  ⇒
| | . . . . . | . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . . | . . |  & (∀e':E.
| | . . . . . | . . |     loc(e') = loc(e)
| | . . . . . | . . |     ⇒
| | . . . . . | . . |     ¬(pred(e) <c e' & e' <c e))
| | . . . . . | . . 6. ∀e:E.
| | . . . . . | . . |  ¬first(e)
| | . . . . . | . . |  ⇒
| | . . . . . | . . |  (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . . | . . 7. Trans e,e':E. e <c e'
| | . . . . . | . . 8. SWellFounded(e <c e')
| | . . . . . | . . 9. ∀e:E.
| | . . . . . | . . |  isrcv(kind(e))
| | . . . . . | . . |  ⇒
| | . . . . . | . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . . | . . |  =
| | . . . . . | . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . . | . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . . | . . 11. ∀e,e':E.
| | . . . . . | . . |   e <c e'
| | . . . . . | . . |   ⇒
| | . . . . . | . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . . | . . |   ∨ isrcv(kind(e'))
| | . . . . . | . . |     & e <c sender(e') ∨ e = sender(e')
| | . . . . . | . . 12. ∀e:E.
| | . . . . . | . . |   isrcv(kind(e))
| | . . . . . | . . |   ⇒
| | . . . . . | . . |   loc(e) = destination(lnk(kind(e)))
| | . . . . . | . . 13. ∀e:E, l:IdLnk.
| | . . . . . | . . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . . | . . 14. e : E
| | . . . . . | . . 15. e' : E
| | . . . . . | . . 16. isrcv(kind(e))
| | . . . . . | . . 17. isrcv(kind(e'))
| | . . . . . | . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . . | . . 19. t' : ℕ
| | . . . . . | . . 20. t : ℕ
| | . . . . . | . . 21. l : IdLnk
| | . . . . . | . . 22. lnk(kind(e')) = l
| | . . . . . | . . 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . . | . . 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . . | . . |   +||onlnk(l;m(source(l);t))||
| | . . . . . | . . 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . . | . . 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . . | . . |   +||onlnk(l;m(source(l);t'))||
| | . . . . . | . . 27. loc(e) = loc(e')
| | . . . . . | . . 28. time(e)<time(e')
| | . . . . . | . . 29. ||rcvs(l;time(e))||<||rcvs(l;time(e'))||
| | . . . . . | . . 30. ¬t<t'
| | . . . . . | . . ⊢  source(l) = source(l) & t<t'
| | . . . . . | . . |  ∨ source(l) = source(l) & t = t'
| | . . . . . | . . |    & ||rcvs(l;time(e))||
```

```
| | . . . . . | . . |      -||snds(l;t)||<||rcvs(l;time(e'))||
| | . . . . . | . . |      -||snds(l;t')||
| | . . . . . | . . | by Assert (||snds(l;t')||≤||snds(l;t)||)
| | . . . . . | . . |
| | . . . . . | . . |\
| | . . . . . | . . | .....assertion .....
| | . . . . . | . . | 1. the_w : World
| | . . . . . | . . | 2. FairFifo
| | . . . . . | . . | 3. ∀e,e':E.
| | . . . . . | . . |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . . | . . | 4. ∀e:E.
| | . . . . . | . . |    first(e)
| | . . . . . | . . |    ⟺
| | . . . . . | . . |    (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . . | . . | 5. ∀e:E.
| | . . . . . | . . |    ¬first(e)
| | . . . . . | . . |    ⇒
| | . . . . . | . . |    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . . | . . |    & (∀e':E.
| | . . . . . | . . |       loc(e') = loc(e)
| | . . . . . | . . |       ⇒
| | . . . . . | . . |       ¬(pred(e) <c e' & e' <c e))
| | . . . . . | . . | 6. ∀e:E.
| | . . . . . | . . |    ¬first(e)
| | . . . . . | . . |    ⇒
| | . . . . . | . . |    (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . . | . . | 7. Trans e,e':E. e <c e'
| | . . . . . | . . | 8. SWellFounded(e <c e')
| | . . . . . | . . | 9. ∀e:E.
| | . . . . . | . . |    isrcv(kind(e))
| | . . . . . | . . |    ⇒
| | . . . . . | . . |    sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . . | . . |    =
| | . . . . . | . . |    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . . | . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . . | . . | 11. ∀e,e':E.
| | . . . . . | . . |     e <c e'
| | . . . . . | . . |     ⇒
| | . . . . . | . . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . . | . . |     ∨ isrcv(kind(e'))
| | . . . . . | . . |       & e <c sender(e') ∨ e = sender(e')
| | . . . . . | . . | 12. ∀e:E.
| | . . . . . | . . |     isrcv(kind(e))
| | . . . . . | . . |     ⇒
| | . . . . . | . . |     loc(e) = destination(lnk(kind(e)))
| | . . . . . | . . | 13. ∀e:E, l:IdLnk.
| | . . . . . | . . |     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . . | . . | 14. e : E
| | . . . . . | . . | 15. e' : E
| | . . . . . | . . | 16. isrcv(kind(e))
| | . . . . . | . . | 17. isrcv(kind(e'))
| | . . . . . | . . | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . . | . . | 19. t' : ℕ
| | . . . . . | . . | 20. t : ℕ
| | . . . . . | . . | 21. l : IdLnk
```

```
| | . . . . | . . | 22. lnk(kind(e')) = l
| | . . . . | . . | 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | . . | 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | . . |     +||onlnk(l;m(source(l);t))||
| | . . . . | . . | 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | . . | 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | . . |     +||onlnk(l;m(source(l);t'))||
| | . . . . | . . | 27. loc(e) = loc(e')
| | . . . . | . . | 28. time(e)<time(e')
| | . . . . | . . | 29. ||rcvs(l;time(e))||<||rcvs(l;time(e'))||
| | . . . . | . . | 30. ¬t<t'
| | . . . . | . . | ⊢  ||snds(l;t')||≤||snds(l;t)||
| | . . . . | . . |   by Unfold `w-snds` 0
| | . . . . | . . |      THEN
| | . . . . | . . |      BackThru
| | . . . . | . . |      Thm* ∀l1,l2:T List. l1 ≤ l2 ⇒ ||l1||≤||l2||
| | . . . . | . . |      {iseg_length}
| | . . . . | . . |      THEN
| | . . . . | . . |      BackThru
| | . . . . | . . |      Thm* ∀ll1,ll2:(T List) List.
| | . . . . | . . |           ll1 ≤ ll2 ⇒ concat(ll1) ≤ concat(ll2)
| | . . . . | . . |      {concat_iseg}
| | . . . . | . . |      THEN
| | . . . . | . . |      BackThru
| | . . . . | . . |      Thm* ∀f:(A→B), L1,L2:A List.
| | . . . . | . . |           L1 ≤ L2 ⇒ map(f;L1) ≤ map(f;L2)
| | . . . . | . . |      {iseg_map}
| | . . . . | . . |      THEN
| | . . . . | . . |      BackThru
| | . . . . | . . |      Thm* ∀i,j:ℕ. i≤j ⇒ upto(i) ≤ upto(j)
| | . . . . | . . |      {upto_iseg}
| | . . . . | . . |
| | . . . . | . . .\
| | . . . . | . . . 1. the_w : World
| | . . . . | . . . 2. FairFifo
| | . . . . | . . . 3. ∀e,e':E.
| | . . . . | . . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | . . . 4. ∀e:E.
| | . . . . | . . . |  first(e)
| | . . . . | . . . |  ⟺
| | . . . . | . . . |  (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | . . . 5. ∀e:E.
| | . . . . | . . . |  ¬first(e)
| | . . . . | . . . |  ⇒
| | . . . . | . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | . . . |  & (∀e':E.
| | . . . . | . . . |     loc(e') = loc(e)

| | . . . . | . . . |     ⇒
| | . . . . | . . . |     ¬(pred(e) <c e' & e' <c e))
| | . . . . | . . . 6. ∀e:E.
| | . . . . | . . . |  ¬first(e)
| | . . . . | . . . |  ⇒
| | . . . . | . . . |  (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | . . . 7. Trans e,e':E. e <c e'
| | . . . . | . . . 8. SWellFounded(e <c e')
```

```
| | . . . . . | . . . 9. ∀e:E.
| | . . . . . | . . . |  isrcv(kind(e))
| | . . . . . | . . . |  ⇒
| | . . . . . | . . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . . | . . . |  =
| | . . . . . | . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . . | . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . . | . . . 11. ∀e,e':E.
| | . . . . . | . . . |   e <c e'
| | . . . . . | . . . |   ⇒
| | . . . . . | . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . . | . . . |   ∨ isrcv(kind(e'))
| | . . . . . | . . . |     & e <c sender(e') ∨ e = sender(e')
| | . . . . . | . . . 12. ∀e:E.
| | . . . . . | . . . |   isrcv(kind(e))
| | . . . . . | . . . |   ⇒
| | . . . . . | . . . |   loc(e) = destination(lnk(kind(e)))
| | . . . . . | . . . 13. ∀e:E, l:IdLnk.
| | . . . . . | . . . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . . | . . . 14. e : E
| | . . . . . | . . . 15. e' : E
| | . . . . . | . . . 16. isrcv(kind(e))
| | . . . . . | . . . 17. isrcv(kind(e'))
| | . . . . . | . . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . . | . . . 19. t' : ℕ
| | . . . . . | . . . 20. t : ℕ
| | . . . . . | . . . 21. l : IdLnk
| | . . . . . | . . . 22. lnk(kind(e')) = l
| | . . . . . | . . . 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . . | . . . 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . . | . . . |   +||onlnk(l;m(source(l);t))||
| | . . . . . | . . . 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . . | . . . 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . . | . . . |   +||onlnk(l;m(source(l);t'))||
| | . . . . . | . . . 27. loc(e) = loc(e')
| | . . . . . | . . . 28. time(e)<time(e')
| | . . . . . | . . . 29. ||rcvs(l;time(e))||<||rcvs(l;time(e'))||
| | . . . . . | . . . 30. ¬t<t'
| | . . . . . | . . . 31. ||snds(l;t')||≤||snds(l;t)||
| | . . . . . | . . . ⊢  source(l) = source(l) & t<t'
| | . . . . . | . . . |  ∨ source(l) = source(l) & t = t'
| | . . . . . | . . . |    & ||rcvs(l;time(e))||
| | . . . . . | . . . |      -||snds(l;t)||<||rcvs(l;time(e'))||
| | . . . . . | . . . |      -||snds(l;t')||  by OrRight
| | . . . . . | . . . |
| | . . . . . | . . . ...
| | . . . . . | . . . |
| | . . . . . | . . . .....assertion .....
| | . . . . . | . . . 1. the_w : World
| | . . . . . | . . . 2. FairFifo
| | . . . . . | . . . 3. ∀e,e':E.
| | . . . . . | . . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . . | . . . 4. ∀e:E.
| | . . . . . | . . . |  first(e)
| | . . . . . | . . . |  ⟺
```

```
| | . . . . | . . . |  (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | . . . 5. ∀e:E.
| | . . . . | . . . |  ¬first(e)
| | . . . . | . . . |  ⇒
| | . . . . | . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . | . . . |  & (∀e':E.
| | . . . . | . . . |     loc(e') = loc(e)
| | . . . . | . . . |     ⇒
| | . . . . | . . . |     ¬(pred(e) <c e' & e' <c e))
| | . . . . | . . . 6. ∀e:E.
| | . . . . | . . . |  ¬first(e)
| | . . . . | . . . |  ⇒
| | . . . . | . . . |  (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | . . . 7. Trans e,e':E. e <c e'
| | . . . . | . . . 8. SWellFounded(e <c e')
| | . . . . | . . . 9. ∀e:E.
| | . . . . | . . . |  isrcv(kind(e))
| | . . . . | . . . |  ⇒
| | . . . . | . . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . | . . . |  =
| | . . . . | . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | . . . 11. ∀e,e':E.
| | . . . . | . . . |   e <c e'
| | . . . . | . . . |   ⇒
| | . . . . | . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . | . . . |   ∨ isrcv(kind(e'))
| | . . . . | . . . |     & e <c sender(e') ∨ e = sender(e')
| | . . . . | . . . 12. ∀e:E.
| | . . . . | . . . |   isrcv(kind(e))
| | . . . . | . . . |   ⇒
| | . . . . | . . . |   loc(e) = destination(lnk(kind(e)))
| | . . . . | . . . 13. ∀e:E, l:IdLnk.
| | . . . . | . . . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | . . . 14. e : E
| | . . . . | . . . 15. e' : E
| | . . . . | . . . 16. isrcv(kind(e))

| | . . . . | . . . 17. isrcv(kind(e'))
| | . . . . | . . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | . . . 19. t' : ℕ
| | . . . . | . . . 20. t : ℕ
| | . . . . | . . . 21. l : IdLnk
| | . . . . | . . . 22. lnk(kind(e')) = l
| | . . . . | . . . 23. ||snds(l;t)||≤||rcvs(l;time(e))||
| | . . . . | . . . 24. ||rcvs(l;time(e))||<||snds(l;t)||
| | . . . . | . . . |   +||onlnk(l;m(source(l);t))||
| | . . . . | . . . 25. ||snds(l;t')||≤||rcvs(l;time(e'))||
| | . . . . | . . . 26. ||rcvs(l;time(e'))||<||snds(l;t')||
| | . . . . | . . . |   +||onlnk(l;m(source(l);t'))||
| | . . . . | . . . 27. loc(e) = loc(e')
| | . . . . | . . . 28. time(e)<time(e')
| | . . . . | . . . 29. ||rcvs(l;time(e))||<||rcvs(l;time(e'))||
| | . . . . | . . . 30. ¬t<t'
| | . . . . | . . . 31. ||snds(l;t')||≤||snds(l;t)||
| | . . . . | . . . 32. t'<t
```

```
| | . . . . | . . . ⊢  ||snds(l;t')||
| | . . . . | . . . |  +||onlnk(l;m(source(l);t'))||≤||snds(l;t)||
| | . . . . | . . . | by Lemmaize [-1]
| | . . . . | . . . |
| | . . . . | . . . ⊢  ∀the_w:World, t',t:ℕ, l:IdLnk.
| | . . . . | . . . |  t'<t
| | . . . . | . . . |  ⇒
| | . . . . | . . . |  ||snds(l;t')||
| | . . . . | . . . |  +||onlnk(l;m(source(l);t'))||≤||snds(l;t)||
| | . . . . | . . . | by Auto
| | . . . . | . . . |    THEN
| | . . . . | . . . |    Subst
| | . . . . | . . . |    (||snds(l;t')||+||onlnk(l;m(source(l);t'))||
| | . . . . | . . . |     =
| | . . . . | . . . |     ||snds(l;t') @ onlnk(l;m(source(l);t'))||)
| | . . . . | . . . |    0
| | . . . . | . . . |
| | . . . . | . . . |\
| | . . . . | . . . | .....equality .....
| | . . . . | . . . | 1. the_w : World
| | . . . . | . . . | 2. t' : ℕ
| | . . . . | . . . | 3. t : ℕ
| | . . . . | . . . | 4. l : IdLnk
| | . . . . | . . . | 5. t'<t
| | . . . . | . . . | ⊢  ||snds(l;t')||+||onlnk(l;m(source(l);t'))||
| | . . . . | . . . | |  =
| | . . . . | . . . | |  ||snds(l;t') @ onlnk(l;m(source(l);t'))||
| | . . . . | . . . | | by RWO
| | . . . . | . . . | |    Thm* ∀as,bs:T List.
| | . . . . | . . . | |          ||as @ bs|| = ||as||+||bs|| ∈ ℤ
| | . . . . | . . . | |    {length_append}
| | . . . . | . . . | |    0
| | . . . . | . . . | |    THEN
| | . . . . | . . . | |    Try
| | . . . . | . . . | |    (Fold `member` 0
| | . . . . | . . . | |     THEN
| | . . . . | . . . | |     BackThru
| | . . . . | . . . | |     Thm* ∀w:World, t:ℕ, l:IdLnk, i:Id.
| | . . . . | . . . | |          ||onlnk(l;m(i;t))|| ∈ ℤ
| | . . . . | . . . | |     {w_DASH_onlnk_DASH_m})
| | . . . . | . . . | |
| | . . . . | . . . | .....unproved Inclusion subgoal .....
| | . . . . | . . . | 1. the_w : World
| | . . . . | . . . | 2. t' : ℕ
| | . . . . | . . . | 3. t : ℕ
| | . . . . | . . . | 4. l : IdLnk
| | . . . . | . . . | 5. t'<t
| | . . . . | . . . | 6. m(source(l);t') = m(source(l);t')
| | . . . . | . . . | 7. {m:Msg(the_w.M)
| | . . . . | . . . |     | source(mlnk(m)) = source(l) } List
| | . . . . | . . . | 8. {m:Msg(the_w.M)
| | . . . . | . . . |     | source(mlnk(m)) = source(l) } List
| | . . . . | . . . | 9. u :
| | . . . . | . . . |    {m:Msg(the_w.M)
| | . . . . | . . . |     | source(mlnk(m)) = source(l) }
```

```
| | . . . . | . . . | ⊢  u ∈ Msg  by Unfold `w-Msg` 0
| | . . . . | . . . |
| | . . . . | . . . .\
| | . . . . | . . . . 1. the_w : World
| | . . . . | . . . . 2. t' : ℕ
| | . . . . | . . . . 3. t : ℕ
| | . . . . | . . . . 4. l : IdLnk
| | . . . . | . . . . 5. t'<t
| | . . . . | . . . . ⊢  ||snds(l;t')
| | . . . . | . . . . |  @ onlnk(l;m(source(l);t'))||≤||snds(l;t)||
| | . . . . | . . . . | by Subst
| | . . . . | . . . . |    ((snds(l;t')
| | . . . . | . . . . |     @ onlnk(l;m(source(l);t'))) ~ snds(l;t'
| | . . . . | . . . . |     +1))
| | . . . . | . . . . |    0
| | . . . . | . . . . |
| | . . . . | . . . . |\
| | . . . . | . . . . | .....equality .....
| | . . . . | . . . . | 1. the_w : World
| | . . . . | . . . . | 2. t' : ℕ
| | . . . . | . . . . | 3. t : ℕ
| | . . . . | . . . . | 4. l : IdLnk
| | . . . . | . . . . | 5. t'<t
| | . . . . | . . . . | ⊢  (snds(l;t')
| | . . . . | . . . . |    @ onlnk(l;m(source(l);t'))) ~ snds(l;t'+1)
| | . . . . | . . . . |   by Unfold `w-snds` 0
| | . . . . | . . . . |      THEN
| | . . . . | . . . . |      RW (AddrC [2] (RecUnfoldC `upto`)) 0
| | . . . . | . . . . |      THEN
| | . . . . | . . . . |      SplitOnConclITE
| | . . . . | . . . . |      THEN
| | . . . . | . . . . |      Try (Complete Auto)
| | . . . . | . . . . |      THEN
| | . . . . | . . . . |      Subst' (t'+1-1 = t') 0
| | . . . . | . . . . |      THEN
| | . . . . | . . . . |      RWO
| | . . . . | . . . . |      Thm* ∀f,as':Top, A:Type, as:A List.
| | . . . . | . . . . |           map(f;as @ as') ~ (map(f;as)
| | . . . . | . . . . |           @ map(f;as'))
| | . . . . | . . . . |      {map_append_sq}
| | . . . . | . . . . |      0
| | . . . . | . . . . |      THEN
| | . . . . | . . . . |      Reduce 0
| | . . . . | . . . . |      THEN
| | . . . . | . . . . |      RWO
| | . . . . | . . . . |      Thm* ∀ll1,ll2:(Top List) List.
| | . . . . | . . . . |           concat(ll1 @ ll2) ~ (concat(ll1)
| | . . . . | . . . . |           @ concat(ll2))
| | . . . . | . . . . |      {concat_append}
| | . . . . | . . . . |      0
| | . . . . | . . . . |      THEN
| | . . . . | . . . . |      Analyze
| | . . . . | . . . . |      THEN
| | . . . . | . . . . |      Try Trivial
| | . . . . | . . . . |      THEN
```

```
| | . . . . | . . . . |       RWO
| | . . . . | . . . . |       Thm* ∀l:Top List, ll:(Top List) List.
| | . . . . | . . . . |            concat([l / ll]) ~ (l @ concat(ll))
| | . . . . | . . . . |       {concat_DASH_cons}
| | . . . . | . . . . |       0
| | . . . . | . . . . |       THEN
| | . . . . | . . . . |       Reduce 0
| | . . . . | . . . . |       THEN
| | . . . . | . . . . |       RWO
| | . . . . | . . . . |       Thm* ∀l:T List. (l @ nil) ~ l
| | . . . . | . . . . |       {append_nil_sq}
| | . . . . | . . . . |       0
| | . . . . | . . . . |       THEN
| | . . . . | . . . . |       Unfold `w-ml` 0
| | . . . . | . . . . |
| | . . . . | . . . . .\
| | . . . . | . . . . . 1. the_w : World
| | . . . . | . . . . . 2. t' : ℕ
| | . . . . | . . . . . 3. t : ℕ
| | . . . . | . . . . . 4. l : IdLnk
| | . . . . | . . . . . 5. t'<t
| | . . . . | . . . . . ⊢  ||snds(l;t'+1)||≤||snds(l;t)||
| | . . . . | . . . . .   by BackThru
| | . . . . | . . . . .      Thm* ∀l1,l2:T List.
| | . . . . | . . . . .           l1 ≤ l2 ⇒ ||l1||≤||l2||
| | . . . . | . . . . .      {iseg_length}
| | . . . . | . . . . .      THEN
| | . . . . | . . . . .      Unfold `w-snds` 0
| | . . . . | . . . . .      THEN
| | . . . . | . . . . .      BackThru
| | . . . . | . . . . .      Thm* ∀ll1,ll2:(T List) List.
| | . . . . | . . . . .           ll1 ≤ ll2
| | . . . . | . . . . .           ⇒
| | . . . . | . . . . .           concat(ll1) ≤ concat(ll2)
| | . . . . | . . . . .      {concat_iseg}
| | . . . . | . . . . .      THEN
| | . . . . | . . . . .      BackThru
| | . . . . | . . . . .      Thm* ∀f:(A→B), L1,L2:A List.
| | . . . . | . . . . .           L1 ≤ L2 ⇒ map(f;L1) ≤ map(f;L2)
| | . . . . | . . . . .      {iseg_map}
| | . . . . | . . . . .      THEN
| | . . . . | . . . . .      BackThru
| | . . . . | . . . . .      Thm* ∀i,j:ℕ. i≤j ⇒ upto(i) ≤ upto(j)
| | . . . . | . . . . .      {upto_iseg}
| | . . . . |
| | . . . . |\
| | . . . . | 1. the_w : World
| | . . . . | 2. FairFifo
| | . . . . | 3. ∀e,e':E.
| | . . . . |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . | 4. ∀e:E.
| | . . . . |    first(e) ⇐⇒ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . | 5. ∀e:E.
| | . . . . |    ¬first(e)
| | . . . . |    ⇒
```

```
| | . . . . |    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . |    & (∀e':E.
| | . . . . |       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . | 6. ∀e:E.
| | . . . . |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . | 7. Trans e,e':E. e <c e'
| | . . . . | 8. SWellFounded(e <c e')
| | . . . . | 9. ∀e:E.
| | . . . . |    isrcv(kind(e))
| | . . . . |    ⇒
| | . . . . |    sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . |    =
| | . . . . |    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . | 11. ∀e,e':E.
| | . . . . |     e <c e'
| | . . . . |     ⇒
| | . . . . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . . . | 12. ∀e:E.
| | . . . . |     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . . . | 13. ∀e:E, l:IdLnk.
| | . . . . |     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . | 14. e : E
| | . . . . | 15. e' : E
| | . . . . | 16. isrcv(kind(e))
| | . . . . | 17. isrcv(kind(e'))
| | . . . . | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . | 19. t' : ℕ
| | . . . . | 20. t : ℕ
| | . . . . | 21. l : IdLnk
| | . . . . | 22. lnk(kind(e')) = l
| | . . . . | 23. match(l;t;time(e))
| | . . . . | 24. match(l;t';time(e'))
| | . . . . | 25. source(l) = source(l) & t<t'
| | . . . . |     ∨ source(l) = source(l) & t = t'
| | . . . . |       & ||rcvs(l;time(e))||
| | . . . . |         -||snds(l;t)||<||rcvs(l;time(e'))||
| | . . . . |         -||snds(l;t')||
| | . . . . | ⊢  loc(e) = loc(e')
| | . . . . |   by AssertBY
| | . . . . |      (destination(lnk(kind(e)))
| | . . . . |       =
| | . . . . |       destination(lnk(kind(e'))))
| | . . . . |      Analyze
| | . . . . |      THEN
| | . . . . |      InstHyp [e] 12
| | . . . . |      THEN
| | . . . . |      InstHyp [e'] 12
| | . . . . |
| | . . . . .\
| | . . . . . 1. the_w : World
| | . . . . . 2. FairFifo
| | . . . . . 3. ∀e,e':E.
| | . . . . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
```

```
| | . . . . . . 4. ∀e:E.
| | . . . . . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . . . 5. ∀e:E.
| | . . . . . |  ¬first(e)
| | . . . . . |  ⇒
| | . . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . . |  & (∀e':E.
| | . . . . . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . . . 6. ∀e:E.
| | . . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . . . 7. Trans e,e':E. e <c e'
| | . . . . . . 8. SWellFounded(e <c e')
| | . . . . . . 9. ∀e:E.
| | . . . . . |  isrcv(kind(e))
| | . . . . . |  ⇒
| | . . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . . |  =
| | . . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . . . 11. ∀e,e':E.
| | . . . . . |   e <c e'
| | . . . . . |   ⇒
| | . . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| | . . . . . . 12. ∀e:E.
| | . . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| | . . . . . . 13. ∀e:E, l:IdLnk.
| | . . . . . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . . . 14. e : E
| | . . . . . . 15. e' : E
| | . . . . . . 16. isrcv(kind(e))
| | . . . . . . 17. isrcv(kind(e'))
| | . . . . . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . . . 19. t' : ℕ
| | . . . . . . 20. t : ℕ
| | . . . . . . 21. l : IdLnk
| | . . . . . . 22. lnk(kind(e')) = l
| | . . . . . . 23. match(l;t;time(e))
| | . . . . . . 24. match(l;t';time(e'))
| | . . . . . . 25. source(l) = source(l) & t<t'
| | . . . . . |   ∨ source(l) = source(l) & t = t'
| | . . . . . |     & ||rcvs(l;time(e))||
| | . . . . . |       -||snds(l;t)||<||rcvs(l;time(e'))||
| | . . . . . |       -||snds(l;t')||
| | . . . . . ⊢  time(e)<time(e')  by Analyze -1 THEN ExRepD
| | . . . . . |
| | . . . . . |\
| | . . . . . | ...
| | . . . . . | |
| | . . . . . | |\
| | . . . . . | | .....assertion .....
| | . . . . . | | 1. the_w : World
| | . . . . . | | 2. FairFifo
| | . . . . . | | 3. ∀e,e':E.
| | . . . . . | |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
```

```
| | . . . . . . | | 4. ∀e:E.
| | . . . . . . | |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . . . | | 5. ∀e:E.
| | . . . . . . | |    ¬first(e)
| | . . . . . . | |    ⇒
| | . . . . . . | |    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . . . | |    & (∀e':E.
| | . . . . . . | |       loc(e') = loc(e)
| | . . . . . . | |       ⇒
| | . . . . . . | |       ¬(pred(e) <c e' & e' <c e))
| | . . . . . . | | 6. ∀e:E.
| | . . . . . . | |    ¬first(e)
| | . . . . . . | |    ⇒
| | . . . . . . | |    (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . . . | | 7. Trans e,e':E. e <c e'
| | . . . . . . | | 8. SWellFounded(e <c e')
| | . . . . . . | | 9. ∀e:E.
| | . . . . . . | |    isrcv(kind(e))
| | . . . . . . | |    ⇒
| | . . . . . . | |    sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . . . | |    =
| | . . . . . . | |    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . . . | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . . . | | 11. ∀e,e':E.
| | . . . . . . | |     e <c e'
| | . . . . . . | |     ⇒
| | . . . . . . | |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . . . | |     ∨ isrcv(kind(e'))
| | . . . . . . | |       & e <c sender(e') ∨ e = sender(e')
| | . . . . . . | | 12. ∀e:E.
| | . . . . . . | |     isrcv(kind(e))
| | . . . . . . | |     ⇒
| | . . . . . . | |     loc(e) = destination(lnk(kind(e)))
| | . . . . . . | | 13. ∀e:E, l:IdLnk.
| | . . . . . . | |     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . . . | | 14. e : E
| | . . . . . . | | 15. e' : E
| | . . . . . . | | 16. isrcv(kind(e))
| | . . . . . . | | 17. isrcv(kind(e'))
| | . . . . . . | | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . . . | | 19. t' : ℕ
| | . . . . . . | | 20. t : ℕ
| | . . . . . . | | 21. l : IdLnk
| | . . . . . . | | 22. lnk(kind(e')) = l
| | . . . . . . | | 23. match(l;t;time(e))
| | . . . . . . | | 24. match(l;t';time(e'))
| | . . . . . . | | 25. source(l) = source(l)
| | . . . . . . | | 26. t<t'
| | . . . . . . | | 27. time(e')≤time(e)
| | . . . . . . | | ⊢  ||rcvs(l;time(e'))||≤||rcvs(l;time(e))||
| | . . . . . . | |   by (BackThru
| | . . . . . . | |       Thm* ∀l1,l2:T List. l1 ≤ l2 ⇒ ||l1||≤||l2||
| | . . . . . . | |       {iseg_length})
| | . . . . . . | |      THEN
| | . . . . . . | |      (Unfold `w-rcvs` 0)
```

```
| | . . . . . . | |       THEN
| | . . . . . . | |       (BackThru
| | . . . . . . | |        Thm* ∀P:(T→𝔹), L2,L1:T List.
| | . . . . . . | |             L1 ≤ L2 ⇒ filter(P;L1) ≤ filter(P;L2)
| | . . . . . . | |        {filter_iseg})
| | . . . . . . | |       THEN
| | . . . . . . | |       (BackThru
| | . . . . . . | |        Thm* ∀f:(A→B), L1,L2:A List.
| | . . . . . . | |             L1 ≤ L2 ⇒ map(f;L1) ≤ map(f;L2)
| | . . . . . . | |        {iseg_map})
| | . . . . . . | |       THEN
| | . . . . . . | |       (BackThru
| | . . . . . . | |        Thm* ∀i,j:ℕ. i≤j ⇒ upto(i) ≤ upto(j)
| | . . . . . . | |        {upto_iseg})
| | . . . . . . | |
| | . . . . . . | .\
| | . . . . . . | . 1. the_w : World
| | . . . . . . | . 2. FairFifo
| | . . . . . . | . 3. ∀e,e':E.
| | . . . . . . | . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . . . | . 4. ∀e:E.
| | . . . . . . | . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . . . | . 5. ∀e:E.
| | . . . . . . | . |  ¬first(e)
| | . . . . . . | . |  ⇒
| | . . . . . . | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . . . | . |  & (∀e':E.
| | . . . . . . | . |     loc(e') = loc(e)
| | . . . . . . | . |     ⇒
| | . . . . . . | . |     ¬(pred(e) <c e' & e' <c e))
| | . . . . . . | . 6. ∀e:E.
| | . . . . . . | . |  ¬first(e)
| | . . . . . . | . |  ⇒
| | . . . . . . | . |  (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . . . | . 7. Trans e,e':E. e <c e'
| | . . . . . . | . 8. SWellFounded(e <c e')
| | . . . . . . | . 9. ∀e:E.
| | . . . . . . | . |  isrcv(kind(e))
| | . . . . . . | . |  ⇒
| | . . . . . . | . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . . . | . |  =
| | . . . . . . | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . . . | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . . . | . 11. ∀e,e':E.
| | . . . . . . | . |   e <c e'
| | . . . . . . | . |   ⇒
| | . . . . . . | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . . . | . |   ∨ isrcv(kind(e'))
| | . . . . . . | . |     & e <c sender(e') ∨ e = sender(e')
| | . . . . . . | . 12. ∀e:E.
| | . . . . . . | . |   isrcv(kind(e))
| | . . . . . . | . |   ⇒
| | . . . . . . | . |   loc(e) = destination(lnk(kind(e)))
| | . . . . . . | . 13. ∀e:E, l:IdLnk.
| | . . . . . . | . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
```

```
| | . . . . . . | . 14. e : E
| | . . . . . . | . 15. e' : E
| | . . . . . . | . 16. isrcv(kind(e))
| | . . . . . . | . 17. isrcv(kind(e'))
| | . . . . . . | . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . . . | . 19. t' : ℕ
| | . . . . . . | . 20. t : ℕ
| | . . . . . . | . 21. l : IdLnk
| | . . . . . . | . 22. lnk(kind(e')) = l
| | . . . . . . | . 23. match(l;t;time(e))
| | . . . . . . | . 24. match(l;t';time(e'))
| | . . . . . . | . 25. source(l) = source(l)
| | . . . . . . | . 26. t<t'
| | . . . . . . | . 27. time(e')≤time(e)
| | . . . . . . | . 28. ||rcvs(l;time(e'))||≤||rcvs(l;time(e))||
| | . . . . . . | . ⊢  time(e)<time(e')
| | . . . . . . | . | by Assert
| | . . . . . . | . |    (||snds(l;t)||
| | . . . . . . | . |     +||onlnk(l;m(source(l);t))||≤||snds(l;t')||)
| | . . . . . . | . |
| | . . . . . . | . |\
| | . . . . . . | . | .....assertion .....
| | . . . . . . | . | 1. the_w : World
| | . . . . . . | . | 2. FairFifo
| | . . . . . . | . | 3. ∀e,e':E.
| | . . . . . . | . | |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . . . | . | 4. ∀e:E.
| | . . . . . . | . | |  first(e)
| | . . . . . . | . | |  ⟺
| | . . . . . . | . | |  (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . . . | . | 5. ∀e:E.
| | . . . . . . | . | |  ¬first(e)
| | . . . . . . | . | |  ⇒
| | . . . . . . | . | |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . . . | . | |  & (∀e':E.
| | . . . . . . | . | |     loc(e') = loc(e)
| | . . . . . . | . | |     ⇒
| | . . . . . . | . | |     ¬(pred(e) <c e' & e' <c e))
| | . . . . . . | . | 6. ∀e:E.
| | . . . . . . | . | |  ¬first(e)
| | . . . . . . | . | |  ⇒
| | . . . . . . | . | |  (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . . . | . | 7. Trans e,e':E. e <c e'
| | . . . . . . | . | 8. SWellFounded(e <c e')
| | . . . . . . | . | 9. ∀e:E.
| | . . . . . . | . | |  isrcv(kind(e))
| | . . . . . . | . | |  ⇒
| | . . . . . . | . | |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . . . | . | |  =
| | . . . . . . | . | |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . . . | . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . . . | . | 11. ∀e,e':E.
| | . . . . . . | . | |   e <c e'
| | . . . . . . | . | |   ⇒
| | . . . . . . | . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
```

```
| | . . . . . . | . | |   ∨ isrcv(kind(e'))
| | . . . . . . | . | |     & e <c sender(e') ∨ e = sender(e')
| | . . . . . . | . | 12. ∀e:E.
| | . . . . . . | . | |   isrcv(kind(e))
| | . . . . . . | . | |   ⇒
| | . . . . . . | . | |   loc(e) = destination(lnk(kind(e)))
| | . . . . . . | . | 13. ∀e:E, l:IdLnk.
| | . . . . . . | . | |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . . . | . | 14. e : E
| | . . . . . . | . | 15. e' : E
| | . . . . . . | . | 16. isrcv(kind(e))
| | . . . . . . | . | 17. isrcv(kind(e'))
| | . . . . . . | . | 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . . . | . | 19. t' : ℕ
| | . . . . . . | . | 20. t : ℕ
| | . . . . . . | . | 21. l : IdLnk
| | . . . . . . | . | 22. lnk(kind(e')) = l
| | . . . . . . | . | 23. match(l;t;time(e))
| | . . . . . . | . | 24. match(l;t';time(e'))
| | . . . . . . | . | 25. source(l) = source(l)
| | . . . . . . | . | 26. t<t'
| | . . . . . . | . | 27. time(e')≤time(e)
| | . . . . . . | . | 28. ||rcvs(l;time(e'))||≤||rcvs(l;time(e))||
| | . . . . . . | . | ⊢  ||snds(l;t)||
| | . . . . . . | . | |  +||onlnk(l;m(source(l);t))||≤||snds(l;t')||
| | . . . . . . | . | | by Lemmaize [-3]
| | . . . . . . | . | |
| | . . . . . . | . | ⊢  ∀the_w:World, t',t:ℕ, l:IdLnk.
| | . . . . . . | . | |  t<t'
| | . . . . . . | . | |  ⇒
| | . . . . . . | . | |  ||snds(l;t)||
| | . . . . . . | . | |  +||onlnk(l;m(source(l);t))||≤||snds(l;t')||
| | . . . . . . | . | | by Auto
| | . . . . . . | . | |    THEN
| | . . . . . . | . | |    Subst
| | . . . . . . | . | |    (||snds(l;t)||+||onlnk(l;m(source(l);t))||
| | . . . . . . | . | |     =
| | . . . . . . | . | |     ||snds(l;t) @ onlnk(l;m(source(l);t))||)
| | . . . . . . | . | |    0
| | . . . . . . | . | |    THENA
| | . . . . . . | . | |    (Auto
| | . . . . . . | . | |     THEN

| | . . . . . . | . | |     Try
| | . . . . . . | . | |     (BackThru
| | . . . . . . | . | |      Thm* ∀w:World, t:ℕ, l:IdLnk, i:Id.
| | . . . . . . | . | |           ||onlnk(l;m(i;t))|| ∈ ℤ
| | . . . . . . | . | |      {w_DASH_onlnk_DASH_m}))
| | . . . . . . | . | |
| | . . . . . . | . | |\
| | . . . . . . | . | | .....equality .....
| | . . . . . . | . | | 1. the_w : World
| | . . . . . . | . | | 2. t' : ℕ
| | . . . . . . | . | | 3. t : ℕ
| | . . . . . . | . | | 4. l : IdLnk
| | . . . . . . | . | | 5. t<t'
```

```
| | . . . . . . | . | | ⊢  ||snds(l;t)||+||onlnk(l;m(source(l);t))||
| | . . . . . . | . | |    =
| | . . . . . . | . | |    ||snds(l;t) @ onlnk(l;m(source(l);t))||
| | . . . . . . | . | |   by RWO
| | . . . . . . | . | |      Thm* ∀as,bs:T List.
| | . . . . . . | . | |           ||as @ bs|| = ||as||+||bs|| ∈ ℤ
| | . . . . . . | . | |      {length_append}
| | . . . . . . | . | |      0
| | . . . . . . | . | |      THEN
| | . . . . . . | . | |      Try
| | . . . . . . | . | |      (Fold `member` 0
| | . . . . . . | . | |       THEN
| | . . . . . . | . | |       BackThru
| | . . . . . . | . | |       Thm* ∀w:World, t:ℕ, l:IdLnk, i:Id.
| | . . . . . . | . | |            ||onlnk(l;m(i;t))|| ∈ ℤ
| | . . . . . . | . | |       {w_DASH_onlnk_DASH_m})
| | . . . . . . | . | |      THEN
| | . . . . . . | . | |      Unfold `w-Msg` 0
| | . . . . . . | . | |
| | . . . . . . | . | .\
| | . . . . . . | . | . 1. the_w : World
| | . . . . . . | . | . 2. t' : ℕ
| | . . . . . . | . | . 3. t : ℕ
| | . . . . . . | . | . 4. l : IdLnk
| | . . . . . . | . | . 5. t<t'
| | . . . . . . | . | . ⊢  ||snds(l;t)
| | . . . . . . | . | . |  @ onlnk(l;m(source(l);t))||≤||snds(l;t')||
| | . . . . . . | . | . | by Subst
| | . . . . . . | . | . |    ((snds(l;t)
| | . . . . . . | . | . |     @ onlnk(l;m(source(l);t))) ~ snds(l;t+1))
| | . . . . . . | . | . |    0
| | . . . . . . | . | . |
| | . . . . . . | . | . |\
| | . . . . . . | . | . | .....equality .....
| | . . . . . . | . | . | 1. the_w : World
| | . . . . . . | . | . | 2. t' : ℕ
| | . . . . . . | . | . | 3. t : ℕ
| | . . . . . . | . | . | 4. l : IdLnk
| | . . . . . . | . | . | 5. t<t'
| | . . . . . . | . | . | ⊢  (snds(l;t)
| | . . . . . . | . | . |    @ onlnk(l;m(source(l);t))) ~ snds(l;t+1)
| | . . . . . . | . | . |   by Unfold `w-snds` 0
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      RW (AddrC [2] (RecUnfoldC `upto`)) 0
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      SplitOnConclITE
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      Try (Complete Auto)
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      Subst' (t+1-1 = t) 0
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      RWO
| | . . . . . . | . | . |      Thm* ∀f,as':Top, A:Type, as:A List.
| | . . . . . . | . | . |           map(f;as @ as') ~ (map(f;as)
| | . . . . . . | . | . |           @ map(f;as'))
```

```
| | . . . . . . | . | . |      {map_append_sq}
| | . . . . . . | . | . |      0
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      Reduce 0
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      RWO
| | . . . . . . | . | . |      Thm* ∀ll1,ll2:(Top List) List.
| | . . . . . . | . | . |           concat(ll1 @ ll2) ~ (concat(ll1)
| | . . . . . . | . | . |           @ concat(ll2))
| | . . . . . . | . | . |      {concat_append}
| | . . . . . . | . | . |      0
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      Analyze
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      Try Trivial
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      RWO
| | . . . . . . | . | . |      Thm* ∀l:Top List, ll:(Top List) List.
| | . . . . . . | . | . |           concat([l / ll]) ~ (l @ concat(ll))
| | . . . . . . | . | . |      {concat_DASH_cons}
| | . . . . . . | . | . |      0
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      Reduce 0
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      RWO
| | . . . . . . | . | . |      Thm* ∀l:T List. (l @ nil) ~ l
| | . . . . . . | . | . |      {append_nil_sq}
| | . . . . . . | . | . |      0
| | . . . . . . | . | . |      THEN
| | . . . . . . | . | . |      Unfold `w-ml` 0
| | . . . . . . | . | . |
| | . . . . . . | . | . .\
| | . . . . . . | . | . . 1. the_w : World
| | . . . . . . | . | . . 2. t' : ℕ
| | . . . . . . | . | . . 3. t : ℕ
| | . . . . . . | . | . . 4. l : IdLnk
| | . . . . . . | . | . . 5. t<t'
| | . . . . . . | . | . . ⊢  ||snds(l;t+1)||≤||snds(l;t')||
| | . . . . . . | . | . .   by BackThru
| | . . . . . . | . | . .      Thm* ∀l1,l2:T List.
| | . . . . . . | . | . .           l1 ≤ l2 ⇒ ||l1||≤||l2||
| | . . . . . . | . | . .      {iseg_length}
| | . . . . . . | . | . .      THEN
| | . . . . . . | . | . .      Unfold `w-snds` 0
| | . . . . . . | . | . .      THEN
| | . . . . . . | . | . .      BackThru
| | . . . . . . | . | . .      Thm* ∀ll1,ll2:(T List) List.
| | . . . . . . | . | . .           ll1 ≤ ll2
| | . . . . . . | . | . .           ⇒
| | . . . . . . | . | . .           concat(ll1) ≤ concat(ll2)
| | . . . . . . | . | . .      {concat_iseg}
| | . . . . . . | . | . .      THEN
| | . . . . . . | . | . .      BackThru
| | . . . . . . | . | . .      Thm* ∀f:(A→B), L1,L2:A List.
| | . . . . . . | . | . .           L1 ≤ L2 ⇒ map(f;L1) ≤ map(f;L2)
```

```
| | . . . . . . | . | . .      {iseg_map}
| | . . . . . . | . | . .      THEN
| | . . . . . . | . | . .      BackThru
| | . . . . . . | . | . .      Thm* ∀i,j:ℕ. i≤j ⇒ upto(i) ≤ upto(j)
| | . . . . . . | . | . .      {upto_iseg}
| | . . . . . . | . |
| | . . . . . . | . .\
| | . . . . . . | . . 1. the_w : World
| | . . . . . . | . . 2. FairFifo
| | . . . . . . | . . 3. ∀e,e':E.
| | . . . . . . | . .    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . . . | . . 4. ∀e:E.
| | . . . . . . | . .    first(e)
| | . . . . . . | . .    ⟺
| | . . . . . . | . .    (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . . . | . . 5. ∀e:E.
| | . . . . . . | . .    ¬first(e)
| | . . . . . . | . .    ⇒
| | . . . . . . | . .    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . . . | . .    & (∀e':E.
| | . . . . . . | . .       loc(e') = loc(e)
| | . . . . . . | . .       ⇒
| | . . . . . . | . .       ¬(pred(e) <c e' & e' <c e))
| | . . . . . . | . . 6. ∀e:E.
| | . . . . . . | . .    ¬first(e)
| | . . . . . . | . .    ⇒
| | . . . . . . | . .    (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . . . | . . 7. Trans e,e':E. e <c e'
| | . . . . . . | . . 8. SWellFounded(e <c e')
| | . . . . . . | . . 9. ∀e:E.
| | . . . . . . | . .    isrcv(kind(e))
| | . . . . . . | . .    ⇒
| | . . . . . . | . .    sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . . . | . .    =
| | . . . . . . | . .    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . . . | . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . . . | . . 11. ∀e,e':E.
| | . . . . . . | . .     e <c e'
| | . . . . . . | . .     ⇒
| | . . . . . . | . .     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| | . . . . . . | . .     ∨ isrcv(kind(e'))
| | . . . . . . | . .       & e <c sender(e') ∨ e = sender(e')
| | . . . . . . | . . 12. ∀e:E.
| | . . . . . . | . .     isrcv(kind(e))
| | . . . . . . | . .     ⇒
| | . . . . . . | . .     loc(e) = destination(lnk(kind(e)))
| | . . . . . . | . . 13. ∀e:E, l:IdLnk.
| | . . . . . . | . .     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . . . | . . 14. e : E
| | . . . . . . | . . 15. e' : E
| | . . . . . . | . . 16. isrcv(kind(e))
| | . . . . . . | . . 17. isrcv(kind(e'))
| | . . . . . . | . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . . . | . . 19. t' : ℕ
| | . . . . . . | . . 20. t : ℕ
```

```
| | . . . . . . | . . 21. l : IdLnk
| | . . . . . . | . . 22. lnk(kind(e')) = l
| | . . . . . . | . . 23. match(l;t;time(e))
| | . . . . . . | . . 24. match(l;t';time(e'))
| | . . . . . . | . . 25. source(l) = source(l)
| | . . . . . . | . . 26. t<t'
| | . . . . . . | . . 27. time(e')≤time(e)
| | . . . . . . | . . 28. ||rcvs(l;time(e'))||≤||rcvs(l;time(e))||
| | . . . . . . | . . 29. ||snds(l;t)||
| | . . . . . . | . .     +||onlnk(l;m(source(l);t))||≤||snds(l;t')||
| | . . . . . . | . . ⊢  time(e)<time(e')
| | . . . . . . | . .   by AllHyps
| | . . . . . . | . .      (λi.
| | . . . . . . | . .       RWO
| | . . . . . . | . .       Thm* ∀the_w:World, l:IdLnk, t,t':ℕ.
| | . . . . . . | . .            match(l;t;t')
| | . . . . . . | . .            ⟺
| | . . . . . . | . .            ||snds(l;t)||≤||rcvs(l;t')||
| | . . . . . . | . .            & ||rcvs(l;t')||<||snds(l;t)||
| | . . . . . . | . .              +||onlnk(l;m(source(l);t))||
| | . . . . . . | . .       {assert_DASH_w_DASH_match}
| | . . . . . . | . .       i

| | . . . . . . | . .       THEN
| | . . . . . . | . .       ExRepD)
| | . . . . . . |
| | . . . . . . .\
| | . . . . . . . 1. the_w : World
| | . . . . . . . 2. FairFifo
| | . . . . . . . 3. ∀e,e':E.
| | . . . . . . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . . . . 4. ∀e:E.
| | . . . . . . . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . . . . 5. ∀e:E.
| | . . . . . . . |  ¬first(e)
| | . . . . . . . |  ⇒
| | . . . . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . . . . |  & (∀e':E.
| | . . . . . . . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . . . . 6. ∀e:E.
| | . . . . . . . |  ¬first(e)
| | . . . . . . . |  ⇒
| | . . . . . . . |  (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . . . . 7. Trans e,e':E. e <c e'
| | . . . . . . . 8. SWellFounded(e <c e')
| | . . . . . . . 9. ∀e:E.
| | . . . . . . . |  isrcv(kind(e))
| | . . . . . . . |  ⇒
| | . . . . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . . . . |  =
| | . . . . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . . . . 11. ∀e,e':E.
| | . . . . . . . |   e <c e'
| | . . . . . . . |   ⇒
| | . . . . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
```

```
| | . . . . . . . |   ∨ isrcv(kind(e'))
| | . . . . . . . |     & e <c sender(e') ∨ e = sender(e')
| | . . . . . . . 12. ∀e:E.
| | . . . . . . . |   isrcv(kind(e))
| | . . . . . . . |   ⇒
| | . . . . . . . |   loc(e) = destination(lnk(kind(e)))
| | . . . . . . . 13. ∀e:E, l:IdLnk.
| | . . . . . . . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . . . . 14. e : E
| | . . . . . . . 15. e' : E
| | . . . . . . . 16. isrcv(kind(e))
| | . . . . . . . 17. isrcv(kind(e'))
| | . . . . . . . 18. lnk(kind(e)) = lnk(kind(e'))
| | . . . . . . . 19. t' : ℕ
| | . . . . . . . 20. t : ℕ
| | . . . . . . . 21. l : IdLnk
| | . . . . . . . 22. lnk(kind(e')) = l
| | . . . . . . . 23. match(l;t;time(e))
| | . . . . . . . 24. match(l;t';time(e'))
| | . . . . . . . 25. source(l) = source(l)
| | . . . . . . . 26. t = t'
| | . . . . . . . 27. ||rcvs(l;time(e))||
| | . . . . . . . |   -||snds(l;t)||<||rcvs(l;time(e'))||-||snds(l;t')||
| | . . . . . . . ⊢  time(e)<time(e')
| | . . . . . . . | by HypSubst' -2 -1 THEN Decide (time(e')≤time(e))
| | . . . . . . . |
| | . . . . . . . ...
| | . . . . . . . |
| | . . . . . . . .....assertion .....
| | . . . . . . . 1. the_w : World
| | . . . . . . . 2. FairFifo
| | . . . . . . . 3. ∀e,e':E.
| | . . . . . . .    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| | . . . . . . . 4. ∀e:E.
| | . . . . . . .    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| | . . . . . . . 5. ∀e:E.
| | . . . . . . .    ¬first(e)
| | . . . . . . .    ⇒
| | . . . . . . .    loc(pred(e)) = loc(e) & pred(e) <c e
| | . . . . . . .    & (∀e':E.
| | . . . . . . .       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| | . . . . . . . 6. ∀e:E.
| | . . . . . . .    ¬first(e)
| | . . . . . . .    ⇒
| | . . . . . . .    (∀x:Id. (x when e) = (x after pred(e)))
| | . . . . . . . 7. Trans e,e':E. e <c e'
| | . . . . . . . 8. SWellFounded(e <c e')
| | . . . . . . . 9. ∀e:E.
| | . . . . . . .    isrcv(kind(e))
| | . . . . . . .    ⇒
| | . . . . . . .    sends(lnk(kind(e));sender(e))[index(e)]
| | . . . . . . .    =
| | . . . . . . .    msg(lnk(kind(e));tag(kind(e));val(e))
| | . . . . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| | . . . . . . . 11. ∀e,e':E.
```

```
| | . . . . . .     e <c e’
| | . . . . . .     ⇒
| | . . . . . .     ¬first(e’) & e <c pred(e’) ∨ e = pred(e’)
| | . . . . . .     ∨ isrcv(kind(e’))
| | . . . . . .       & e <c sender(e’) ∨ e = sender(e’)
| | . . . . . . 12. ∀e:E.
| | . . . . . .     isrcv(kind(e))
| | . . . . . .     ⇒
| | . . . . . .     loc(e) = destination(lnk(kind(e)))
| | . . . . . . 13. ∀e:E, l:IdLnk.
| | . . . . . .     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| | . . . . . . 14. e : E

| | . . . . . . 15. e’ : E
| | . . . . . . 16. isrcv(kind(e))
| | . . . . . . 17. isrcv(kind(e’))
| | . . . . . . 18. lnk(kind(e)) = lnk(kind(e’))
| | . . . . . . 19. t’ : ℕ
| | . . . . . . 20. t : ℕ
| | . . . . . . 21. l : IdLnk
| | . . . . . . 22. lnk(kind(e’)) = l
| | . . . . . . 23. match(l;t;time(e))
| | . . . . . . 24. match(l;t’;time(e’))
| | . . . . . . 25. source(l) = source(l)
| | . . . . . . 26. t = t’
| | . . . . . . 27. ||rcvs(l;time(e))||
| | . . . . . .     -||snds(l;t’)||<||rcvs(l;time(e’))||
| | . . . . . .     -||snds(l;t’)||
| | . . . . . . 28. time(e’)≤time(e)
| | . . . . . . ⊢  ||rcvs(l;time(e’))||≤||rcvs(l;time(e))||
| | . . . . . .   by (BackThru
| | . . . . . .       Thm* ∀l1,l2:T List. l1 ≤ l2 ⇒ ||l1||≤||l2||
| | . . . . . .       {iseg_length})
| | . . . . . .      THEN
| | . . . . . .      (Unfold ‘w-rcvs‘ 0)
| | . . . . . .      THEN
| | . . . . . .      (BackThru
| | . . . . . .       Thm* ∀P:(T→𝔹), L2,L1:T List.
| | . . . . . .            L1 ≤ L2 ⇒ filter(P;L1) ≤ filter(P;L2)
| | . . . . . .       {filter_iseg})
| | . . . . . .      THEN
| | . . . . . .      (BackThru
| | . . . . . .       Thm* ∀f:(A→B), L1,L2:A List.
| | . . . . . .            L1 ≤ L2 ⇒ map(f;L1) ≤ map(f;L2)
| | . . . . . .       {iseg_map})
| | . . . . . .      THEN
| | . . . . . .      (BackThru
| | . . . . . .       Thm* ∀i,j:ℕ. i≤j ⇒ upto(i) ≤ upto(j)
| | . . . . . .       {upto_iseg})
| |
| .\
| . 1. the_w : World
| . 2. FairFifo
| . 3. ∀e,e’:E. loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
| . 4. ∀e:E. first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
| . 5. ∀e:E.
```

```
| . |  ¬first(e)
| . |  ⇒
| . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . 7. Trans e,e':E. e <c e'
| . 8. SWellFounded(e <c e')
| . 9. ∀e:E.
| . |  isrcv(kind(e))
| . |  ⇒
| . |  sends(lnk(kind(e));sender(e))[index(e)]
| . |  =
| . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . 11. ∀e,e':E.
| . |   e <c e'
| . |   ⇒
| . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . 14. ∀e,e':E.
| . |   isrcv(kind(e))
| . |   ⇒
| . |   isrcv(kind(e'))
| . |   ⇒
| . |   lnk(kind(e)) = lnk(kind(e'))
| . |   ⇒
| . |   (e <loc e'
| . |    ⟺
| . |    sender(e) <loc sender(e')
| . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . ⊢  ∀e,e':E.
| . |  isrcv(kind(e))
| . |  ⇒
| . |  isrcv(kind(e'))
| . |  ⇒
| . |  lnk(kind(e)) = lnk(kind(e'))
| . |  ⇒
| . |  (e <c e'
| . |   ⟺
| . |   sender(e) <c sender(e')
| . |   ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | by RepeatFor 6 (ParallelOp -1)
| . |
| . |\
| . | .....antecedent .....
| . | 1. the_w : World
| . | 2. FairFifo
| . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | 5. ∀e:E.
| . | |  ¬first(e)
| . | |  ⇒
| . | |  loc(pred(e)) = loc(e) & pred(e) <c e
```

```
| . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | 7. Trans e,e':E. e <c e'
| . | 8. SWellFounded(e <c e')
| . | 9. ∀e:E.
| . | |  isrcv(kind(e))
| . | |  ⇒
| . | |  sends(lnk(kind(e));sender(e))[index(e)]
| . | |  =
| . | |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | 11. ∀e,e':E.
| . | |   e <c e'
| . | |   ⇒
| . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | 14. ∀e,e':E.
| . | |   isrcv(kind(e))
| . | |   ⇒
| . | |   isrcv(kind(e'))
| . | |   ⇒
| . | |   lnk(kind(e)) = lnk(kind(e'))
| . | |   ⇒
| . | |   (e <loc e'
| . | |    ⟺
| . | |    sender(e) <loc sender(e')
| . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | 15. e : E
| . | 16. ∀e':E.
| . | |   isrcv(kind(e))
| . | |   ⇒
| . | |   isrcv(kind(e'))
| . | |   ⇒
| . | |   lnk(kind(e)) = lnk(kind(e'))
| . | |   ⇒
| . | |   (e <loc e'
| . | |    ⟺
| . | |    sender(e) <loc sender(e')
| . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | 17. e' : E
| . | 18. isrcv(kind(e))
| . | 19. isrcv(kind(e'))
| . | 20. lnk(kind(e)) = lnk(kind(e'))
| . | 21. e <c e'
| . | ⊢  e <loc e'
| . | | by RWO
| . | |    Thm* ∀the_w:World, e,e':E.
| . | |         FairFifo
| . | |         ⇒
| . | |         (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| . | |    {w_DASH_locl_DASH_iff}
| . | |    0
| . | |
```

```
| . | 1. the_w : World
| . | 2. FairFifo
| . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | 5. ∀e:E.
| . |    ¬first(e)
| . |    ⇒
| . |    loc(pred(e)) = loc(e) & pred(e) <c e
| . |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | 7. Trans e,e':E. e <c e'
| . | 8. SWellFounded(e <c e')
| . | 9. ∀e:E.
| . |    isrcv(kind(e))
| . |    ⇒
| . |    sends(lnk(kind(e));sender(e))[index(e)]
| . |    =
| . |    msg(lnk(kind(e));tag(kind(e));val(e))
| . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | 11. ∀e,e':E.
| . |     e <c e'
| . |     ⇒
| . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | 14. ∀e,e':E.
| . |     isrcv(kind(e))
| . |     ⇒
| . |     isrcv(kind(e'))
| . |     ⇒
| . |     lnk(kind(e)) = lnk(kind(e'))
| . |     ⇒
| . |     (e <loc e'
| . |      ⟺
| . |      sender(e) <loc sender(e')
| . |      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | 15. e : E
| . | 16. ∀e':E.
| . |     isrcv(kind(e))
| . |     ⇒
| . |     isrcv(kind(e'))
| . |     ⇒
| . |     lnk(kind(e)) = lnk(kind(e'))
| . |     ⇒
| . |     (e <loc e'
| . |      ⟺

| . |      sender(e) <loc sender(e')
| . |      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | 17. e' : E
| . | 18. isrcv(kind(e))
| . | 19. isrcv(kind(e'))
| . | 20. lnk(kind(e)) = lnk(kind(e'))
| . | 21. e <c e'
| . | ⊢  loc(e) = loc(e')
```

```
| . |   by AssertBY (loc(e) = destination(lnk(kind(e))))
| . |      BackThruSomeHyp
| . |      THEN
| . |      AssertBY (loc(e') = destination(lnk(kind(e'))))
| . |      BackThruSomeHyp
| . |      THEN
| . |      Subst (lnk(kind(e)) = lnk(kind(e'))) -2
| . |
| . |\
| . | 1. the_w : World
| . | 2. FairFifo
| . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | 5. ∀e:E.
| . |    ¬first(e)
| . |    ⇒
| . |    loc(pred(e)) = loc(e) & pred(e) <c e
| . |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | 7. Trans e,e':E. e <c e'
| . | 8. SWellFounded(e <c e')
| . | 9. ∀e:E.
| . |    isrcv(kind(e))
| . |    ⇒
| . |    sends(lnk(kind(e));sender(e))[index(e)]
| . |    =
| . |    msg(lnk(kind(e));tag(kind(e));val(e))
| . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | 11. ∀e,e':E.
| . |     e <c e'
| . |     ⇒
| . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | 14. ∀e,e':E.
| . |     isrcv(kind(e))
| . |     ⇒
| . |     isrcv(kind(e'))
| . |     ⇒
| . |     lnk(kind(e)) = lnk(kind(e'))
| . |     ⇒
| . |     (e <loc e'
| . |      ⟺
| . |      sender(e) <loc sender(e')
| . |      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | 15. e : E
| . | 16. ∀e':E.
| . |     isrcv(kind(e))
| . |     ⇒
| . |     isrcv(kind(e'))
| . |     ⇒
| . |     lnk(kind(e)) = lnk(kind(e'))
| . |     ⇒
| . |     (e <loc e'
```

```
| . |      ⟺
| . |      sender(e) <loc sender(e')
| . |      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | 17. e' : E
| . | 18. isrcv(kind(e))
| . | 19. isrcv(kind(e'))
| . | 20. lnk(kind(e)) = lnk(kind(e'))
| . | 21. e <c e'
| . | 22. sender(e) <loc sender(e')
| . |     ∨ sender(e) = sender(e') & index(e)<index(e')
| . | ⊢  sender(e) <c sender(e')
| . |    ∨ sender(e) = sender(e') & index(e)<index(e')
| . |   by ParallelOp -1
| . |      THEN
| . |      RWO
| . |      Thm* ∀the_w:World, e,e':E.
| . |           FairFifo
| . |           ⇒
| . |           (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| . |      {w_DASH_locl_DASH_iff}
| . |      -1
| . |
| . |\
| . | .....antecedent .....
| . | 1. the_w : World
| . | 2. FairFifo
| . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | 5. ∀e:E.
| . | |  ¬first(e)
| . | |  ⇒
| . | |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | 7. Trans e,e':E. e <c e'
| . | 8. SWellFounded(e <c e')
| . | 9. ∀e:E.
| . | |  isrcv(kind(e))
| . | |  ⇒
| . | |  sends(lnk(kind(e));sender(e))[index(e)]
| . | |  =
| . | |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | 11. ∀e,e':E.
| . | |   e <c e'
| . | |   ⇒
| . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | 14. ∀e,e':E.
| . | |   isrcv(kind(e))
| . | |   ⇒
| . | |   isrcv(kind(e'))
| . | |   ⇒
```

```
| . | |   lnk(kind(e)) = lnk(kind(e’))
| . | |   ⇒
| . | |   (e <loc e’
| . | |    ⟺
| . | |    sender(e) <loc sender(e’)
| . | |    ∨ sender(e) = sender(e’) & index(e)<index(e’))
| . | 15. e : E
| . | 16. ∀e’:E.
| . | |   isrcv(kind(e))
| . | |   ⇒
| . | |   isrcv(kind(e’))
| . | |   ⇒
| . | |   lnk(kind(e)) = lnk(kind(e’))
| . | |   ⇒
| . | |   (e <loc e’
| . | |    ⟺
| . | |    sender(e) <loc sender(e’)
| . | |    ∨ sender(e) = sender(e’) & index(e)<index(e’))
| . | 17. e’ : E
| . | 18. isrcv(kind(e))
| . | 19. isrcv(kind(e’))
| . | 20. lnk(kind(e)) = lnk(kind(e’))
| . | 21. sender(e) <c sender(e’)
| . | |   ∨ sender(e) = sender(e’) & index(e)<index(e’)
| . | ⊢  sender(e) <loc sender(e’)
| . | |  ∨ sender(e) = sender(e’) & index(e)<index(e’)
| . | | by ParallelOp -1
| . | |    THEN
| . | |    RWO
| . | |    Thm* ∀the_w:World, e,e’:E.
| . | |         FairFifo
| . | |         ⇒
| . | |         (e <loc e’ ⟺ loc(e) = loc(e’) ∈ Id & e <c e’)
| . | |    {w_DASH_locl_DASH_iff}
| . | |    0
| . | |
| . | 1. the_w : World
| . | 2. FairFifo
| . | 3. ∀e,e’:E. loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
| . | 4. ∀e:E. first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
| . | 5. ∀e:E.
| . | |  ¬first(e)
| . | |  ⇒
| . | |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | |  & (∀e’:E. loc(e’) = loc(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
| . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | 7. Trans e,e’:E. e <c e’
| . | 8. SWellFounded(e <c e’)
| . | 9. ∀e:E.
| . | |  isrcv(kind(e))
| . | |  ⇒
| . | |  sends(lnk(kind(e));sender(e))[index(e)]
| . | |  =
| . | |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
```

```
| . | 11. ∀e,e':E.
| . | |   e <c e'
| . | |   ⇒
| . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | 14. ∀e,e':E.
| . | |   isrcv(kind(e))
| . | |   ⇒
| . | |   isrcv(kind(e'))
| . | |   ⇒
| . | |   lnk(kind(e)) = lnk(kind(e'))
| . | |   ⇒
| . | |   (e <loc e'
| . | |    ⟺
| . | |    sender(e) <loc sender(e')
| . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | 15. e : E
| . | 16. ∀e':E.
| . | |   isrcv(kind(e))
| . | |   ⇒
| . | |   isrcv(kind(e'))
| . | |   ⇒
| . | |   lnk(kind(e)) = lnk(kind(e'))
| . | |   ⇒
| . | |   (e <loc e'
| . | |    ⟺
| . | |    sender(e) <loc sender(e')
| . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | 17. e' : E
| . | 18. isrcv(kind(e))
| . | 19. isrcv(kind(e'))
| . | 20. lnk(kind(e)) = lnk(kind(e'))
| . | 21. sender(e) <c sender(e')
| . | ⊢  loc(sender(e)) = loc(sender(e'))
| . | | by Assert
| . | |    (∀e:E.
| . | |     isrcv(kind(e)) ⇒ loc(sender(e)) = source(lnk(kind(e))))
| . | |
| . | |\
| . | | .....assertion .....
| . | | 1. the_w : World
| . | | 2. FairFifo
| . | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | | 5. ∀e:E.
| . | | |  ¬first(e)
| . | | |  ⇒
| . | | |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | | 7. Trans e,e':E. e <c e'
| . | | 8. SWellFounded(e <c e')
| . | | 9. ∀e:E.
```

```
| . | | |  isrcv(kind(e))
| . | | |  ⇒
| . | | |  sends(lnk(kind(e));sender(e))[index(e)]
| . | | |  =
| . | | |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | | 11. ∀e,e':E.
| . | | |   e <c e'
| . | | |   ⇒
| . | | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | | 14. ∀e,e':E.
| . | | |   isrcv(kind(e))
| . | | |   ⇒
| . | | |   isrcv(kind(e'))
| . | | |   ⇒
| . | | |   lnk(kind(e)) = lnk(kind(e'))
| . | | |   ⇒
| . | | |   (e <loc e'
| . | | |    ⟺
| . | | |    sender(e) <loc sender(e')
| . | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | 15. e : E
| . | | 16. ∀e':E.
| . | | |   isrcv(kind(e))
| . | | |   ⇒
| . | | |   isrcv(kind(e'))
| . | | |   ⇒
| . | | |   lnk(kind(e)) = lnk(kind(e'))
| . | | |   ⇒
| . | | |   (e <loc e'
| . | | |    ⟺
| . | | |    sender(e) <loc sender(e')
| . | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | 17. e' : E
| . | | 18. isrcv(kind(e))
| . | | 19. isrcv(kind(e'))
| . | | 20. lnk(kind(e)) = lnk(kind(e'))
| . | | 21. sender(e) <c sender(e')
| . | | ⊢  ∀e:E.
| . | | |  isrcv(kind(e)) ⇒ loc(sender(e)) = source(lnk(kind(e)))
| . | | | by Auto
| . | | |    THEN
| . | | |    Assert Dec(loc(sender(e1)) = source(lnk(kind(e1))))
| . | | |
| . | | |\
| . | | | .....assertion .....
| . | | | 1. the_w : World
| . | | | 2. FairFifo
| . | | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | | | 5. ∀e:E.
| . | | | |  ¬first(e)
```

```
| . | | | |  ⇒
| . | | | |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | | | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | | | 6. ∀e:E.
| . | | | |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | | | 7. Trans e,e':E. e <c e'
| . | | | 8. SWellFounded(e <c e')
| . | | | 9. ∀e:E.
| . | | | |  isrcv(kind(e))
| . | | | |  ⇒
| . | | | |  sends(lnk(kind(e));sender(e))[index(e)]
| . | | | |  =
| . | | | |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | | | 11. ∀e,e':E.
| . | | | |   e <c e'
| . | | | |   ⇒
| . | | | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | | | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | | | 12. ∀e:E.
| . | | | |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | | | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | | | 14. ∀e,e':E.
| . | | | |   isrcv(kind(e))
| . | | | |   ⇒
| . | | | |   isrcv(kind(e'))
| . | | | |   ⇒
| . | | | |   lnk(kind(e)) = lnk(kind(e'))
| . | | | |   ⇒
| . | | | |   (e <loc e'
| . | | | |    ⟺
| . | | | |    sender(e) <loc sender(e')
| . | | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | | 15. e : E
| . | | | 16. ∀e':E.
| . | | | |   isrcv(kind(e))
| . | | | |   ⇒
| . | | | |   isrcv(kind(e'))
| . | | | |   ⇒
| . | | | |   lnk(kind(e)) = lnk(kind(e'))
| . | | | |   ⇒
| . | | | |   (e <loc e'
| . | | | |    ⟺
| . | | | |    sender(e) <loc sender(e')
| . | | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | | 17. e' : E
| . | | | 18. isrcv(kind(e))
| . | | | 19. isrcv(kind(e'))
| . | | | 20. lnk(kind(e)) = lnk(kind(e'))
| . | | | 21. sender(e) <c sender(e')
| . | | | 22. e1 : E
| . | | | 23. isrcv(kind(e1))
| . | | | ⊢  Dec(loc(sender(e1)) = source(lnk(kind(e1))))
| . | | | | by Assert (∀i,j:Id. Dec(i = j))
| . | | | |
```

```
| . | | | |\
| . | | | | .....assertion .....
| . | | | | 1. the_w : World
| . | | | | 2. FairFifo
| . | | | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | | | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | | | | 5. ∀e:E.
| . | | | |    ¬first(e)
| . | | | |    ⇒
| . | | | |    loc(pred(e)) = loc(e) & pred(e) <c e
| . | | | |    & (∀e':E.
| . | | | |       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | | | | 6. ∀e:E.
| . | | | |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | | | | 7. Trans e,e':E. e <c e'
| . | | | | 8. SWellFounded(e <c e')
| . | | | | 9. ∀e:E.
| . | | | |    isrcv(kind(e))
| . | | | |    ⇒
| . | | | |    sends(lnk(kind(e));sender(e))[index(e)]
| . | | | |    =
| . | | | |    msg(lnk(kind(e));tag(kind(e));val(e))
| . | | | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | | | | 11. ∀e,e':E.
| . | | | |     e <c e'
| . | | | |     ⇒
| . | | | |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | | | |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | | | | 12. ∀e:E.
| . | | | |     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | | | | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | | | | 14. ∀e,e':E.
| . | | | |     isrcv(kind(e))
| . | | | |     ⇒
| . | | | |     isrcv(kind(e'))
| . | | | |     ⇒
| . | | | |     lnk(kind(e)) = lnk(kind(e'))
| . | | | |     ⇒
| . | | | |     (e <loc e'
| . | | | |      ⟺
| . | | | |      sender(e) <loc sender(e')
| . | | | |      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | | | 15. e : E
| . | | | | 16. ∀e':E.
| . | | | |     isrcv(kind(e))
| . | | | |     ⇒
| . | | | |     isrcv(kind(e'))
| . | | | |     ⇒
| . | | | |     lnk(kind(e)) = lnk(kind(e'))
| . | | | |     ⇒
| . | | | |     (e <loc e'
| . | | | |      ⟺
| . | | | |      sender(e) <loc sender(e')
| . | | | |      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | | | 17. e' : E
```

```
| . | | | | | 18. isrcv(kind(e))
| . | | | | | 19. isrcv(kind(e'))
| . | | | | | 20. lnk(kind(e)) = lnk(kind(e'))
| . | | | | | 21. sender(e) <c sender(e')
| . | | | | | 22. e1 : E
| . | | | | | 23. isrcv(kind(e1))
| . | | | | | ⊢  ∀i,j:Id. Dec(i = j)  by Auto
| . | | | | |
| . | | | | .\
| . | | | | . 1. the_w : World
| . | | | | . 2. FairFifo
| . | | | | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | | | | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | | | | . 5. ∀e:E.
| . | | | | .    ¬first(e)
| . | | | | .    ⇒
| . | | | | .    loc(pred(e)) = loc(e) & pred(e) <c e
| . | | | | .    & (∀e':E.
| . | | | | .       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | | | | . 6. ∀e:E.
| . | | | | .    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | | | | . 7. Trans e,e':E. e <c e'
| . | | | | . 8. SWellFounded(e <c e')
| . | | | | . 9. ∀e:E.
| . | | | | .    isrcv(kind(e))
| . | | | | .    ⇒
| . | | | | .    sends(lnk(kind(e));sender(e))[index(e)]
| . | | | | .    =
| . | | | | .    msg(lnk(kind(e));tag(kind(e));val(e))
| . | | | | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | | | | . 11. ∀e,e':E.
| . | | | | .     e <c e'
| . | | | | .     ⇒
| . | | | | .     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | | | | .     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | | | | . 12. ∀e:E.
| . | | | | .     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | | | | . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | | | | . 14. ∀e,e':E.
| . | | | | .     isrcv(kind(e))
| . | | | | .     ⇒
| . | | | | .     isrcv(kind(e'))
| . | | | | .     ⇒
| . | | | | .     lnk(kind(e)) = lnk(kind(e'))
| . | | | | .     ⇒
| . | | | | .     (e <loc e'
| . | | | | .      ⟺
| . | | | | .      sender(e) <loc sender(e')
| . | | | | .      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | | | . 15. e : E
| . | | | | . 16. ∀e':E.
| . | | | | .     isrcv(kind(e))
| . | | | | .     ⇒
| . | | | | .     isrcv(kind(e'))
| . | | | | .     ⇒
```

```
| . | | | .     lnk(kind(e)) = lnk(kind(e'))
| . | | | .     ⇒
| . | | | .     (e <loc e'
| . | | | .      ⟺
| . | | | .      sender(e) <loc sender(e')
| . | | | .      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | | . 17. e' : E
| . | | | . 18. isrcv(kind(e))
| . | | | . 19. isrcv(kind(e'))
| . | | | . 20. lnk(kind(e)) = lnk(kind(e'))
| . | | | . 21. sender(e) <c sender(e')
| . | | | . 22. e1 : E
| . | | | . 23. isrcv(kind(e1))
| . | | | . 24. ∀i,j:Id. Dec(i = j)
| . | | | . ⊢  Dec(loc(sender(e1)) = source(lnk(kind(e1))))
| . | | | .   by BackThru -1
| . | | |
| . | | .\
| . | | . 1. the_w : World
| . | | . 2. FairFifo
| . | | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | | . 5. ∀e:E.
| . | | . |  ¬first(e)
| . | | . |  ⇒
| . | | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | | . 6. ∀e:E.
| . | | . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | | . 7. Trans e,e':E. e <c e'
| . | | . 8. SWellFounded(e <c e')
| . | | . 9. ∀e:E.
| . | | . |  isrcv(kind(e))
| . | | . |  ⇒
| . | | . |  sends(lnk(kind(e));sender(e))[index(e)]
| . | | . |  =
| . | | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | | . 11. ∀e,e':E.
| . | | . |   e <c e'
| . | | . |   ⇒
| . | | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | | . 12. ∀e:E.
| . | | . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | | . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | | . 14. ∀e,e':E.
| . | | . |   isrcv(kind(e))
| . | | . |   ⇒
| . | | . |   isrcv(kind(e'))
| . | | . |   ⇒
| . | | . |   lnk(kind(e)) = lnk(kind(e'))

| . | | . |   ⇒
| . | | . |   (e <loc e'
| . | | . |    ⟺
```

```
| . | | . |    sender(e) <loc sender(e')
| . | | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | . 15. e : E
| . | | . 16. ∀e':E.
| . | | . |   isrcv(kind(e))
| . | | . |   ⇒
| . | | . |   isrcv(kind(e'))
| . | | . |   ⇒
| . | | . |   lnk(kind(e)) = lnk(kind(e'))
| . | | . |   ⇒
| . | | . |   (e <loc e'
| . | | . |    ⟺
| . | | . |    sender(e) <loc sender(e')
| . | | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | . 17. e' : E
| . | | . 18. isrcv(kind(e))
| . | | . 19. isrcv(kind(e'))
| . | | . 20. lnk(kind(e)) = lnk(kind(e'))
| . | | . 21. sender(e) <c sender(e')
| . | | . 22. e1 : E
| . | | . 23. isrcv(kind(e1))
| . | | . 24. Dec(loc(sender(e1)) = source(lnk(kind(e1))))
| . | | . ⊢  loc(sender(e1)) = source(lnk(kind(e1)))  by Analyze -1
| . | | . |
| . | | . 1. the_w : World
| . | | . 2. FairFifo
| . | | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | | . 5. ∀e:E.
| . | | . |  ¬first(e)
| . | | . |  ⇒
| . | | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | | . 6. ∀e:E.
| . | | . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | | . 7. Trans e,e':E. e <c e'
| . | | . 8. SWellFounded(e <c e')
| . | | . 9. ∀e:E.
| . | | . |  isrcv(kind(e))
| . | | . |  ⇒
| . | | . |  sends(lnk(kind(e));sender(e))[index(e)]
| . | | . |  =
| . | | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | | . 11. ∀e,e':E.
| . | | . |   e <c e'
| . | | . |   ⇒
| . | | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | | . 12. ∀e:E.
| . | | . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | | . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | | . 14. ∀e,e':E.
| . | | . |   isrcv(kind(e))
| . | | . |   ⇒
```

```
| . | | . |   isrcv(kind(e'))
| . | | . |   ⇒
| . | | . |   lnk(kind(e)) = lnk(kind(e'))
| . | | . |   ⇒
| . | | . |   (e <loc e'
| . | | . |    ⟺
| . | | . |    sender(e) <loc sender(e')
| . | | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | . 15. e : E
| . | | . 16. ∀e':E.
| . | | . |   isrcv(kind(e))
| . | | . |   ⇒
| . | | . |   isrcv(kind(e'))
| . | | . |   ⇒
| . | | . |   lnk(kind(e)) = lnk(kind(e'))
| . | | . |   ⇒
| . | | . |   (e <loc e'
| . | | . |    ⟺
| . | | . |    sender(e) <loc sender(e')
| . | | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | . 17. e' : E
| . | | . 18. isrcv(kind(e))
| . | | . 19. isrcv(kind(e'))
| . | | . 20. lnk(kind(e)) = lnk(kind(e'))
| . | | . 21. sender(e) <c sender(e')
| . | | . 22. e1 : E
| . | | . 23. isrcv(kind(e1))
| . | | . 24. ¬loc(sender(e1)) = source(lnk(kind(e1)))
| . | | . ⊢  loc(sender(e1)) = source(lnk(kind(e1)))
| . | | . | by AssertBY (sends(lnk(kind(e1));sender(e1)) = nil)
| . | | . |    BackThruSomeHyp
| . | | . |
| . | | . 1. the_w : World
| . | | . 2. FairFifo
| . | | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | | . 5. ∀e:E.
| . | | . |  ¬first(e)
| . | | . |  ⇒
| . | | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | | . 6. ∀e:E.
| . | | . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | | . 7. Trans e,e':E. e <c e'
| . | | . 8. SWellFounded(e <c e')
| . | | . 9. ∀e:E.
| . | | . |  isrcv(kind(e))
| . | | . |  ⇒
| . | | . |  sends(lnk(kind(e));sender(e))[index(e)]
| . | | . |  =
| . | | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | | . 11. ∀e,e':E.
| . | | . |   e <c e'
| . | | . |   ⇒
```

```
| . | | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | | . 12. ∀e:E.
| . | | . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | | . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | | . 14. ∀e,e':E.
| . | | . |   isrcv(kind(e))
| . | | . |   ⇒
| . | | . |   isrcv(kind(e'))
| . | | . |   ⇒
| . | | . |   lnk(kind(e)) = lnk(kind(e'))
| . | | . |   ⇒
| . | | . |   (e <loc e'
| . | | . |    ⟺
| . | | . |    sender(e) <loc sender(e')
| . | | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | . 15. e : E
| . | | . 16. ∀e':E.
| . | | . |   isrcv(kind(e))
| . | | . |   ⇒
| . | | . |   isrcv(kind(e'))
| . | | . |   ⇒
| . | | . |   lnk(kind(e)) = lnk(kind(e'))
| . | | . |   ⇒
| . | | . |   (e <loc e'
| . | | . |    ⟺
| . | | . |    sender(e) <loc sender(e')
| . | | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | . 17. e' : E
| . | | . 18. isrcv(kind(e))
| . | | . 19. isrcv(kind(e'))
| . | | . 20. lnk(kind(e)) = lnk(kind(e'))
| . | | . 21. sender(e) <c sender(e')
| . | | . 22. e1 : E
| . | | . 23. isrcv(kind(e1))
| . | | . 24. ¬loc(sender(e1)) = source(lnk(kind(e1)))
| . | | . 25. sends(lnk(kind(e1));sender(e1)) = nil
| . | | . ⊢  loc(sender(e1)) = source(lnk(kind(e1)))
| . | | . | by AssertBY
| . | | . |    (index(e1) ∈ ℕ||sends(lnk(kind(e1));sender(e1))||)
| . | | . |    Auto
| . | | . |
| . | | . 1. the_w : World
| . | | . 2. FairFifo
| . | | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | | . 5. ∀e:E.
| . | | . |  ¬first(e)
| . | | . |  ⇒
| . | | . |  loc(pred(e)) = loc(e) & pred(e) <c e
| . | | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | | . 6. ∀e:E.
| . | | . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | | . 7. Trans e,e':E. e <c e'
| . | | . 8. SWellFounded(e <c e')
```

```
| . | | . 9. ∀e:E.
| . | | . |  isrcv(kind(e))
| . | | . |  ⇒
| . | | . |  sends(lnk(kind(e));sender(e))[index(e)]
| . | | . |  =
| . | | . |  msg(lnk(kind(e));tag(kind(e));val(e))
| . | | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | | . 11. ∀e,e':E.
| . | | . |   e <c e'
| . | | . |   ⇒
| . | | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | | . 12. ∀e:E.
| . | | . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | | . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | | . 14. ∀e,e':E.
| . | | . |   isrcv(kind(e))
| . | | . |   ⇒
| . | | . |   isrcv(kind(e'))
| . | | . |   ⇒
| . | | . |   lnk(kind(e)) = lnk(kind(e'))
| . | | . |   ⇒
| . | | . |   (e <loc e'
| . | | . |    ⟺
| . | | . |    sender(e) <loc sender(e')
| . | | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | . 15. e : E
| . | | . 16. ∀e':E.
| . | | . |   isrcv(kind(e))
| . | | . |   ⇒
| . | | . |   isrcv(kind(e'))
| . | | . |   ⇒
| . | | . |   lnk(kind(e)) = lnk(kind(e'))
| . | | . |   ⇒
| . | | . |   (e <loc e'
| . | | . |    ⟺
| . | | . |    sender(e) <loc sender(e')
| . | | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | . 17. e' : E
| . | | . 18. isrcv(kind(e))
| . | | . 19. isrcv(kind(e'))
| . | | . 20. lnk(kind(e)) = lnk(kind(e'))
| . | | . 21. sender(e) <c sender(e')
| . | | . 22. e1 : E
| . | | . 23. isrcv(kind(e1))
| . | | . 24. ¬loc(sender(e1)) = source(lnk(kind(e1)))
| . | | . 25. sends(lnk(kind(e1));sender(e1)) = nil
| . | | . 26. index(e1) ∈ ℕ||sends(lnk(kind(e1));sender(e1))||
| . | | . ⊢  loc(sender(e1)) = source(lnk(kind(e1)))
| . | | . | by Subst' (||sends(lnk(kind(e1));sender(e1))|| = 0) -1
| . | | . |
| . | | . |\
| . | | . | 1. the_w : World
| . | | . | 2. FairFifo
| . | | . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
```

```
| . | | . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | | . | 5. ∀e:E.
| . | | . |    ¬first(e)
| . | | . |    ⇒
| . | | . |    loc(pred(e)) = loc(e) & pred(e) <c e
| . | | . |    & (∀e':E.
| . | | . |       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | | . | 6. ∀e:E.
| . | | . |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | | . | 7. Trans e,e':E. e <c e'
| . | | . | 8. SWellFounded(e <c e')
| . | | . | 9. ∀e:E.
| . | | . |    isrcv(kind(e))
| . | | . |    ⇒
| . | | . |    sends(lnk(kind(e));sender(e))[index(e)]
| . | | . |    =
| . | | . |    msg(lnk(kind(e));tag(kind(e));val(e))
| . | | . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | | . | 11. ∀e,e':E.
| . | | . |     e <c e'
| . | | . |     ⇒
| . | | . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | | . |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | | . | 12. ∀e:E.
| . | | . |     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | | . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | | . | 14. ∀e,e':E.
| . | | . |     isrcv(kind(e))
| . | | . |     ⇒
| . | | . |     isrcv(kind(e'))
| . | | . |     ⇒
| . | | . |     lnk(kind(e)) = lnk(kind(e'))
| . | | . |     ⇒
| . | | . |     (e <loc e'
| . | | . |      ⟺
| . | | . |      sender(e) <loc sender(e')
| . | | . |      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | . | 15. e : E
| . | | . | 16. ∀e':E.
| . | | . |     isrcv(kind(e))
| . | | . |     ⇒
| . | | . |     isrcv(kind(e'))
| . | | . |     ⇒
| . | | . |     lnk(kind(e)) = lnk(kind(e'))
| . | | . |     ⇒
| . | | . |     (e <loc e'
| . | | . |      ⟺
| . | | . |      sender(e) <loc sender(e')
| . | | . |      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | . | 17. e' : E
| . | | . | 18. isrcv(kind(e))
| . | | . | 19. isrcv(kind(e'))
| . | | . | 20. lnk(kind(e)) = lnk(kind(e'))
| . | | . | 21. sender(e) <c sender(e')
| . | | . | 22. e1 : E
```

```
| . | | . | 23. isrcv(kind(e1))
| . | | . | 24. ¬loc(sender(e1)) = source(lnk(kind(e1)))
| . | | . | 25. sends(lnk(kind(e1));sender(e1)) = nil
| . | | . | 26. index(e1) ∈ ℕ||sends(lnk(kind(e1));sender(e1))||
| . | | . | ⊢  ||sends(lnk(kind(e1));sender(e1))|| = 0
| . | | . |   by HypSubst -2 0 THEN Reduce 0
| . | | . |
| . | | . .\
| . | | . . 1. the_w : World
| . | | . . 2. FairFifo
| . | | . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | | . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | | . . 5. ∀e:E.
| . | | . .    ¬first(e)
| . | | . .    ⇒
| . | | . .    loc(pred(e)) = loc(e) & pred(e) <c e
| . | | . .    & (∀e':E.
| . | | . .       loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | | . . 6. ∀e:E.
| . | | . .    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | | . . 7. Trans e,e':E. e <c e'
| . | | . . 8. SWellFounded(e <c e')
| . | | . . 9. ∀e:E.
| . | | . .    isrcv(kind(e))
| . | | . .    ⇒
| . | | . .    sends(lnk(kind(e));sender(e))[index(e)]
| . | | . .    =
| . | | . .    msg(lnk(kind(e));tag(kind(e));val(e))
| . | | . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | | . . 11. ∀e,e':E.
| . | | . .     e <c e'
| . | | . .     ⇒
| . | | . .     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | | . .     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | | . . 12. ∀e:E.
| . | | . .     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | | . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | | . . 14. ∀e,e':E.
| . | | . .     isrcv(kind(e))
| . | | . .     ⇒
| . | | . .     isrcv(kind(e'))
| . | | . .     ⇒
| . | | . .     lnk(kind(e)) = lnk(kind(e'))
| . | | . .     ⇒
| . | | . .     (e <loc e'
| . | | . .      ⟺
| . | | . .      sender(e) <loc sender(e')
| . | | . .      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | . . 15. e : E
| . | | . . 16. ∀e':E.
| . | | . .     isrcv(kind(e))
| . | | . .     ⇒
| . | | . .     isrcv(kind(e'))
| . | | . .     ⇒
| . | | . .     lnk(kind(e)) = lnk(kind(e'))
```

```
| . | | . .     ⇒
| . | | . .     (e <loc e'
| . | | . .      ⟺
| . | | . .      sender(e) <loc sender(e')
| . | | . .      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | | . . 17. e' : E
| . | | . . 18. isrcv(kind(e))
| . | | . . 19. isrcv(kind(e'))
| . | | . . 20. lnk(kind(e)) = lnk(kind(e'))
| . | | . . 21. sender(e) <c sender(e')
| . | | . . 22. e1 : E
| . | | . . 23. isrcv(kind(e1))
| . | | . . 24. ¬loc(sender(e1)) = source(lnk(kind(e1)))
| . | | . . 25. sends(lnk(kind(e1));sender(e1)) = nil
| . | | . . 26. index(e1) ∈ ℕ0
| . | | . . ⊢  loc(sender(e1)) = source(lnk(kind(e1)))  by Auto
| . | |
| . | .\
| . | . 1. the_w : World
| . | . 2. FairFifo
| . | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . | . 5. ∀e:E.
| . | .    ¬first(e)
| . | .    ⇒
| . | .    loc(pred(e)) = loc(e) & pred(e) <c e
| . | .    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . | . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . | . 7. Trans e,e':E. e <c e'
| . | . 8. SWellFounded(e <c e')
| . | . 9. ∀e:E.
| . | .    isrcv(kind(e))
| . | .    ⇒
| . | .    sends(lnk(kind(e));sender(e))[index(e)]
| . | .    =
| . | .    msg(lnk(kind(e));tag(kind(e));val(e))
| . | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . | . 11. ∀e,e':E.
| . | .     e <c e'
| . | .     ⇒
| . | .     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . | .     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . | . 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . | . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . | . 14. ∀e,e':E.
| . | .     isrcv(kind(e))
| . | .     ⇒
| . | .     isrcv(kind(e'))
| . | .     ⇒
| . | .     lnk(kind(e)) = lnk(kind(e'))
| . | .     ⇒
| . | .     (e <loc e'
| . | .      ⟺
| . | .      sender(e) <loc sender(e')
| . | .      ∨ sender(e) = sender(e') & index(e)<index(e'))
```

```
| . | . 15. e : E
| . | . 16. ∀e':E.
| . | .     isrcv(kind(e))
| . | .     ⇒
| . | .     isrcv(kind(e'))
| . | .     ⇒
| . | .     lnk(kind(e)) = lnk(kind(e'))
| . | .     ⇒
| . | .     (e <loc e'
| . | .      ⟺
| . | .      sender(e) <loc sender(e')
| . | .      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . | . 17. e' : E
| . | . 18. isrcv(kind(e))
| . | . 19. isrcv(kind(e'))
| . | . 20. lnk(kind(e)) = lnk(kind(e'))
| . | . 21. sender(e) <c sender(e')
| . | . 22. ∀e:E.
| . | .     isrcv(kind(e)) ⇒ loc(sender(e)) = source(lnk(kind(e)))
| . | . ⊢  loc(sender(e)) = loc(sender(e'))
| . | .   by InstHyp [e] -1 THEN InstHyp [e'] -2
| . | .      THEN
| . | .      Subst (lnk(kind(e)) = lnk(kind(e'))) -2
| . |
| . .\
| . . 1. the_w : World
| . . 2. FairFifo
| . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
| . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
| . . 5. ∀e:E.
| . .    ¬first(e)
| . .    ⇒
| . .    loc(pred(e)) = loc(e) & pred(e) <c e
| . .    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
| . . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
| . . 7. Trans e,e':E. e <c e'
| . . 8. SWellFounded(e <c e')
| . . 9. ∀e:E.
| . .    isrcv(kind(e))
| . .    ⇒
| . .    sends(lnk(kind(e));sender(e))[index(e)]
| . .    =
| . .    msg(lnk(kind(e));tag(kind(e));val(e))
| . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
| . . 11. ∀e,e':E.
| . .     e <c e'
| . .     ⇒
| . .     ¬first(e') & e <c pred(e') ∨ e = pred(e')
| . .     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
| . . 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
| . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
| . . 14. ∀e,e':E.
| . .     isrcv(kind(e))
| . .     ⇒
| . .     isrcv(kind(e'))
```

```
| . .     ⇒
| . .     lnk(kind(e)) = lnk(kind(e'))
| . .     ⇒
| . .     (e <loc e'
| . .      ⟺
| . .      sender(e) <loc sender(e')
| . .      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . . 15. e : E
| . . 16. ∀e':E.
| . .     isrcv(kind(e))
| . .     ⇒
| . .     isrcv(kind(e'))
| . .     ⇒
| . .     lnk(kind(e)) = lnk(kind(e'))
| . .     ⇒
| . .     (e <loc e'
| . .      ⟺
| . .      sender(e) <loc sender(e')
| . .      ∨ sender(e) = sender(e') & index(e)<index(e'))
| . . 17. e' : E
| . . 18. isrcv(kind(e))
| . . 19. isrcv(kind(e'))
| . . 20. lnk(kind(e)) = lnk(kind(e'))
| . . 21. sender(e) <c sender(e')
| . .     ∨ sender(e) = sender(e') & index(e)<index(e')
| . . 22. e <loc e'
| . . ⊢  e <c e'
| . .   by RWO
| . .      Thm* ∀the_w:World, e,e':E.
| . .           FairFifo
| . .           ⇒
| . .           (e <loc e' ⟺ loc(e) = loc(e') ∈ Id & e <c e')
| . .      {w_DASH_locl_DASH_iff}
| . .      -1
|
.\
. 1. the_w : World
. 2. FairFifo
. 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. 5. ∀e:E.
. |  ¬first(e)
. |  ⇒
. |  loc(pred(e)) = loc(e) & pred(e) <c e
. |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. 7. Trans e,e':E. e <c e'
. 8. SWellFounded(e <c e')
. 9. ∀e:E.
. |  isrcv(kind(e))
. |  ⇒
. |  sends(lnk(kind(e));sender(e))[index(e)]
. |  =
. |  msg(lnk(kind(e));tag(kind(e));val(e))
. 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
```

```
. 11. ∀e,e':E.
. |   e <c e'
. |   ⇒
. |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. 14. ∀e,e':E.
. |   isrcv(kind(e))
. |   ⇒
. |   isrcv(kind(e'))
. |   ⇒
. |   lnk(kind(e)) = lnk(kind(e'))
. |   ⇒
. |   (e <c e'
. |    ⟺
. |    sender(e) <c sender(e')
. |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. ⊢  ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
. |  ∃e':E.
. |  isrcv(kind(e'))
. |  & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. | by Assert
. |    (∀m:ℕ, l:IdLnk, t:ℕ.
. |     m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||))
. |
. |\
. | .....assertion .....
. | 1. the_w : World
. | 2. FairFifo
. | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | 5. ∀e:E.
. | |  ¬first(e)
. | |  ⇒
. | |  loc(pred(e)) = loc(e) & pred(e) <c e
. | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | 7. Trans e,e':E. e <c e'
. | 8. SWellFounded(e <c e')
. | 9. ∀e:E.
. | |  isrcv(kind(e))
. | |  ⇒
. | |  sends(lnk(kind(e));sender(e))[index(e)]
. | |  =
. | |  msg(lnk(kind(e));tag(kind(e));val(e))
. | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | 11. ∀e,e':E.
. | |   e <c e'
. | |   ⇒
. | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | 14. ∀e,e':E.
```

```
. | |   isrcv(kind(e))
. | |   ⇒
. | |   isrcv(kind(e'))
. | |   ⇒
. | |   lnk(kind(e)) = lnk(kind(e'))
. | |   ⇒
. | |   (e <c e'
. | |    ⟺
. | |    sender(e) <c sender(e')
. | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | ⊢  ∀m:ℕ, l:IdLnk, t:ℕ.
. | |  m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. | | by InductionOnNat
. | |
. | |\
. | | .....basecase .....
. | | 1. the_w : World
. | | 2. FairFifo
. | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | | 5. ∀e:E.
. | |    ¬first(e)
. | |    ⇒
. | |    loc(pred(e)) = loc(e) & pred(e) <c e
. | |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | | 7. Trans e,e':E. e <c e'
. | | 8. SWellFounded(e <c e')
. | | 9. ∀e:E.
. | |    isrcv(kind(e))
. | |    ⇒
. | |    sends(lnk(kind(e));sender(e))[index(e)]
. | |    =
. | |    msg(lnk(kind(e));tag(kind(e));val(e))
. | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | | 11. ∀e,e':E.
. | |     e <c e'
. | |     ⇒
. | |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | | 14. ∀e,e':E.
. | |     isrcv(kind(e))
. | |     ⇒
. | |     isrcv(kind(e'))
. | |     ⇒
. | |     lnk(kind(e)) = lnk(kind(e'))
. | |     ⇒
. | |     (e <c e'
. | |      ⟺
. | |      sender(e) <c sender(e')
. | |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. | | ⊢  ∀l:IdLnk, t:ℕ.
. | |    0≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & 0≤||rcvs(l;t')||)
```

```
. | |   by Auto THEN InstConcl [t]
. | |
. | .\
. | . .....upcase .....
. | . 1. the_w : World
. | . 2. FairFifo
. | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . 5. ∀e:E.
. | . |  ¬first(e)
. | . |  ⇒
. | . |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . 7. Trans e,e':E. e <c e'
. | . 8. SWellFounded(e <c e')
. | . 9. ∀e:E.
. | . |  isrcv(kind(e))
. | . |  ⇒
. | . |  sends(lnk(kind(e));sender(e))[index(e)]
. | . |  =
. | . |  msg(lnk(kind(e));tag(kind(e));val(e))
. | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . 11. ∀e,e':E.
. | . |   e <c e'
. | . |   ⇒
. | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . 14. ∀e,e':E.
. | . |   isrcv(kind(e))
. | . |   ⇒
. | . |   isrcv(kind(e'))
. | . |   ⇒
. | . |   lnk(kind(e)) = lnk(kind(e'))
. | . |   ⇒
. | . |   (e <c e'
. | . |    ⟺
. | . |    sender(e) <c sender(e')
. | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . 15. m : ℤ
. | . 16. 0<m
. | . 17. ∀l:IdLnk, t:ℕ.
. | . |   m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . ⊢  ∀l:IdLnk, t:ℕ.
. | . |  m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. | . | by Auto THEN InstHyp [l;t] -4 THEN ExRepD
. | . |
. | . 1. the_w : World
. | . 2. FairFifo
. | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . 5. ∀e:E.
. | . |  ¬first(e)
```

```
. | . |  ⇒
. | . |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . 7. Trans e,e':E. e <c e'
. | . 8. SWellFounded(e <c e')
. | . 9. ∀e:E.
. | . |  isrcv(kind(e))
. | . |  ⇒
. | . |  sends(lnk(kind(e));sender(e))[index(e)]
. | . |  =
. | . |  msg(lnk(kind(e));tag(kind(e));val(e))
. | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . 11. ∀e,e':E.
. | . |   e <c e'
. | . |   ⇒
. | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . 14. ∀e,e':E.
. | . |   isrcv(kind(e))
. | . |   ⇒
. | . |   isrcv(kind(e'))
. | . |   ⇒
. | . |   lnk(kind(e)) = lnk(kind(e'))
. | . |   ⇒
. | . |   (e <c e'
. | . |    ⟺
. | . |    sender(e) <c sender(e')
. | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . 15. m : ℤ
. | . 16. 0<m
. | . 17. ∀l:IdLnk, t:ℕ.
. | . |   m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . 18. l : IdLnk
. | . 19. t : ℕ
. | . 20. m≤||snds(l;t)||
. | . 21. t' : ℕ
. | . 22. t≤t'
. | . 23. m-1≤||rcvs(l;t')||
. | . ⊢  ∃t':ℕ. t≤t' & m≤||rcvs(l;t')||
. | . | by Analyze 2 THEN Analyze 3 THEN Analyze 4
. | . |    THEN
. | . |    InstHyp [l;t'] 5
. | . |    THEN
. | . |    ExRepD
. | . |
. | . 1. the_w : World
. | . 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . 3. ∀i:Id, t:ℕ.
. | . |  isnull(a(i;t))
. | . |  ⇒
. | . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . 4. ∀i:Id, t:ℕ, l:IdLnk.
```

```
. | . |  isrcv(l;a(i;t))
. | . |  ⇒
. | . |  destination(l) = i
. | . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . 5. ∀l:IdLnk, t:ℕ.
. | . |  ∃t':ℕ.
. | . |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . 8. ∀e:E.
. | . |  ¬first(e)
. | . |  ⇒
. | . |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . 10. Trans e,e':E. e <c e'
. | . 11. SWellFounded(e <c e')
. | . 12. ∀e:E.
. | . |   isrcv(kind(e))
. | . |   ⇒
. | . |   sends(lnk(kind(e));sender(e))[index(e)]
. | . |   =
. | . |   msg(lnk(kind(e));tag(kind(e));val(e))
. | . 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . 14. ∀e,e':E.
. | . |   e <c e'
. | . |   ⇒
. | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . 15. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . 17. ∀e,e':E.
. | . |   isrcv(kind(e))
. | . |   ⇒
. | . |   isrcv(kind(e'))
. | . |   ⇒
. | . |   lnk(kind(e)) = lnk(kind(e'))
. | . |   ⇒
. | . |   (e <c e'
. | . |    ⟺
. | . |    sender(e) <c sender(e')
. | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . 18. m : ℤ
. | . 19. 0<m
. | . 20. ∀l:IdLnk, t:ℕ.
. | . |   m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . 21. l : IdLnk
. | . 22. t : ℕ
. | . 23. m≤||snds(l;t)||
. | . 24. t' : ℕ
. | . 25. t≤t'
. | . 26. m-1≤||rcvs(l;t')||
. | . 27. t'@0 : ℕ
. | . 28. t'≤t'@0
. | . 29. isrcv(l;a(destination(l);t'@0)) ∨ queue(l;t'@0) = nil
```

```
. | . ⊢  ∃t':ℕ. t≤t' & m≤||rcvs(l;t')||  by Analyze -1
. | . |
. | . |\
. | . | 1. the_w : World
. | . | 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . | |  ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . | 3. ∀i:Id, t:ℕ.
. | . | |  isnull(a(i;t))
. | . | |  ⇒
. | . | |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . | 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . | |  isrcv(l;a(i;t))
. | . | |  ⇒
. | . | |  destination(l) = i
. | . | |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . | 5. ∀l:IdLnk, t:ℕ.
. | . | |  ∃t':ℕ.
. | . | |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . | 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . | 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . | 8. ∀e:E.
. | . | |  ¬first(e)
. | . | |  ⇒
. | . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . | 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . | 10. Trans e,e':E. e <c e'
. | . | 11. SWellFounded(e <c e')
. | . | 12. ∀e:E.
. | . | |   isrcv(kind(e))
. | . | |   ⇒
. | . | |   sends(lnk(kind(e));sender(e))[index(e)]
. | . | |   =
. | . | |   msg(lnk(kind(e));tag(kind(e));val(e))
. | . | 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . | 14. ∀e,e':E.
. | . | |   e <c e'
. | . | |   ⇒
. | . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . | 15. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . | 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . | 17. ∀e,e':E.
. | . | |   isrcv(kind(e))
. | . | |   ⇒
. | . | |   isrcv(kind(e'))
. | . | |   ⇒
. | . | |   lnk(kind(e)) = lnk(kind(e'))
. | . | |   ⇒
. | . | |   (e <c e'
. | . | |    ⟺
. | . | |    sender(e) <c sender(e')
. | . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . | 18. m : ℤ
. | . | 19. 0<m
```

```
. | . | 20. ∀l:IdLnk, t:ℕ.
. | . | |   m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . | 21. l : IdLnk
. | . | 22. t : ℕ
. | . | 23. m≤||snds(l;t)||
. | . | 24. t' : ℕ
. | . | 25. t≤t'
. | . | 26. m-1≤||rcvs(l;t')||
. | . | 27. t'@0 : ℕ
. | . | 28. t'≤t'@0
. | . | 29. isrcv(l;a(destination(l);t'@0))
. | . | ⊢  ∃t':ℕ. t≤t' & m≤||rcvs(l;t')||  by InstConcl [t'@0+1]
. | . | |
. | . | 1. the_w : World
. | . | 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . | |  ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . | 3. ∀i:Id, t:ℕ.
. | . | |  isnull(a(i;t))
. | . | |  ⇒
. | . | |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . | 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . | |  isrcv(l;a(i;t))
. | . | |  ⇒
. | . | |  destination(l) = i
. | . | |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . | 5. ∀l:IdLnk, t:ℕ.
. | . | |  ∃t':ℕ.
. | . | |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . | 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . | 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . | 8. ∀e:E.
. | . | |  ¬first(e)
. | . | |  ⇒
. | . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . | 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . | 10. Trans e,e':E. e <c e'
. | . | 11. SWellFounded(e <c e')
. | . | 12. ∀e:E.
. | . | |   isrcv(kind(e))
. | . | |   ⇒
. | . | |   sends(lnk(kind(e));sender(e))[index(e)]
. | . | |   =
. | . | |   msg(lnk(kind(e));tag(kind(e));val(e))
. | . | 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . | 14. ∀e,e':E.
. | . | |   e <c e'
. | . | |   ⇒
. | . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . | 15. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . | 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . | 17. ∀e,e':E.
. | . | |   isrcv(kind(e))
. | . | |   ⇒
```

```
. | . | |   isrcv(kind(e'))
. | . | |   ⇒
. | . | |   lnk(kind(e)) = lnk(kind(e'))
. | . | |   ⇒
. | . | |   (e <c e'
. | . | |    ⟺
. | . | |    sender(e) <c sender(e')
. | . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . | 18. m : ℤ
. | . | 19. 0<m
. | . | 20. ∀l:IdLnk, t:ℕ.
. | . | |   m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . | 21. l : IdLnk
. | . | 22. t : ℕ
. | . | 23. m≤||snds(l;t)||
. | . | 24. t' : ℕ
. | . | 25. t≤t'
. | . | 26. m-1≤||rcvs(l;t')||
. | . | 27. t'@0 : ℕ
. | . | 28. t'≤t'@0
. | . | 29. isrcv(l;a(destination(l);t'@0))
. | . | ⊢  m≤||rcvs(l;t'@0+1)||
. | . | | by Unfold `w-rcvs` 0 THEN RecUnfold `upto` 0
. | . | |    THEN
. | . | |    SplitOnConclITE
. | . | |
. | . | |\
. | . | | .....truecase .....
. | . | | 1. the_w : World
. | . | | 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . | |    ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . | | 3. ∀i:Id, t:ℕ.
. | . | |    isnull(a(i;t))
. | . | |    ⇒
. | . | |    (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . | | 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . | |    isrcv(l;a(i;t))
. | . | |    ⇒
. | . | |    destination(l) = i
. | . | |    & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . | | 5. ∀l:IdLnk, t:ℕ.
. | . | |    ∃t':ℕ.
. | . | |    t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . | | 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . | | 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . | | 8. ∀e:E.
. | . | |    ¬first(e)
. | . | |    ⇒
. | . | |    loc(pred(e)) = loc(e) & pred(e) <c e
. | . | |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . | | 9. ∀e:E.
. | . | |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . | | 10. Trans e,e':E. e <c e'
. | . | | 11. SWellFounded(e <c e')
. | . | | 12. ∀e:E.
```

```
. | . | |     isrcv(kind(e))
. | . | |     ⇒
. | . | |     sends(lnk(kind(e));sender(e))[index(e)]
. | . | |     =
. | . | |     msg(lnk(kind(e));tag(kind(e));val(e))
. | . | | 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . | | 14. ∀e,e':E.
. | . | |     e <c e'
. | . | |     ⇒
. | . | |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . | |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . | | 15. ∀e:E.
. | . | |     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . | | 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . | | 17. ∀e,e':E.
. | . | |     isrcv(kind(e))
. | . | |     ⇒
. | . | |     isrcv(kind(e'))
. | . | |     ⇒
. | . | |     lnk(kind(e)) = lnk(kind(e'))
. | . | |     ⇒
. | . | |     (e <c e'
. | . | |      ⟺
. | . | |      sender(e) <c sender(e')
. | . | |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . | | 18. m : ℤ
. | . | | 19. 0<m
. | . | | 20. ∀l:IdLnk, t:ℕ.
. | . | |     m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . | | 21. l : IdLnk
. | . | | 22. t : ℕ
. | . | | 23. m≤||snds(l;t)||
. | . | | 24. t' : ℕ
. | . | | 25. t≤t'
. | . | | 26. m-1≤||rcvs(l;t')||
. | . | | 27. t'@0 : ℕ
. | . | | 28. t'≤t'@0
. | . | | 29. isrcv(l;a(destination(l);t'@0))
. | . | | 30. t'@0+1 = 0
. | . | | ⊢  m≤||filter(λa.isrcv(l;a);map(λt1.
. | . | |    a(destination(l);t1);nil))||  by Assert False
. | . | |
. | . | .\
. | . | . .....falsecase .....
. | . | . 1. the_w : World
. | . | . 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . | . |  ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . | . 3. ∀i:Id, t:ℕ.
. | . | . |  isnull(a(i;t))
. | . | . |  ⇒
. | . | . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . | . 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . | . |  isrcv(l;a(i;t))
. | . | . |  ⇒
. | . | . |  destination(l) = i
```

```
. | . | . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . | . 5. ∀l:IdLnk, t:ℕ.
. | . | . |  ∃t':ℕ.
. | . | . |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . | . 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . | . 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . | . 8. ∀e:E.
. | . | . |  ¬first(e)
. | . | . |  ⇒
. | . | . |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . | . 9. ∀e:E.
. | . | . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . | . 10. Trans e,e':E. e <c e'
. | . | . 11. SWellFounded(e <c e')
. | . | . 12. ∀e:E.
. | . | . |   isrcv(kind(e))
. | . | . |   ⇒
. | . | . |   sends(lnk(kind(e));sender(e))[index(e)]
. | . | . |   =
. | . | . |   msg(lnk(kind(e));tag(kind(e));val(e))
. | . | . 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . | . 14. ∀e,e':E.
. | . | . |   e <c e'
. | . | . |   ⇒
. | . | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . | . 15. ∀e:E.
. | . | . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . | . 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . | . 17. ∀e,e':E.
. | . | . |   isrcv(kind(e))
. | . | . |   ⇒
. | . | . |   isrcv(kind(e'))
. | . | . |   ⇒
. | . | . |   lnk(kind(e)) = lnk(kind(e'))
. | . | . |   ⇒
. | . | . |   (e <c e'
. | . | . |    ⟺
. | . | . |    sender(e) <c sender(e')
. | . | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . | . 18. m : ℤ
. | . | . 19. 0<m
. | . | . 20. ∀l:IdLnk, t:ℕ.
. | . | . |   m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . | . 21. l : IdLnk
. | . | . 22. t : ℕ
. | . | . 23. m≤||snds(l;t)||
. | . | . 24. t' : ℕ
. | . | . 25. t≤t'
. | . | . 26. m-1≤||rcvs(l;t')||
. | . | . 27. t'@0 : ℕ
. | . | . 28. t'≤t'@0
. | . | . 29. isrcv(l;a(destination(l);t'@0))
. | . | . 30. ¬t'@0+1 = 0
```

```
. | . | . ⊢  m≤||filter(λa.isrcv(l;a);map(λt1.
. | . | . |  a(destination(l);t1);upto(t'@0+1-1) @ [(t'@0+1-1)]))||
. | . | . | by Subst' (t'@0+1-1 = t'@0) 0
. | . | . |
. | . | . 1. the_w : World
. | . | . 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . | . |  ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . | . 3. ∀i:Id, t:ℕ.
. | . | . |  isnull(a(i;t))
. | . | . |  ⇒
. | . | . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . | . 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . | . |  isrcv(l;a(i;t))
. | . | . |  ⇒
. | . | . |  destination(l) = i
. | . | . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . | . 5. ∀l:IdLnk, t:ℕ.
. | . | . |  ∃t':ℕ.
. | . | . |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . | . 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . | . 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . | . 8. ∀e:E.
. | . | . |  ¬first(e)
. | . | . |  ⇒
. | . | . |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . | . 9. ∀e:E.
. | . | . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . | . 10. Trans e,e':E. e <c e'
. | . | . 11. SWellFounded(e <c e')
. | . | . 12. ∀e:E.
. | . | . |   isrcv(kind(e))
. | . | . |   ⇒
. | . | . |   sends(lnk(kind(e));sender(e))[index(e)]
. | . | . |   =
. | . | . |   msg(lnk(kind(e));tag(kind(e));val(e))
. | . | . 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . | . 14. ∀e,e':E.
. | . | . |   e <c e'
. | . | . |   ⇒
. | . | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . | . 15. ∀e:E.
. | . | . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . | . 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . | . 17. ∀e,e':E.
. | . | . |   isrcv(kind(e))
. | . | . |   ⇒
. | . | . |   isrcv(kind(e'))
. | . | . |   ⇒
. | . | . |   lnk(kind(e)) = lnk(kind(e'))
. | . | . |   ⇒
. | . | . |   (e <c e'
. | . | . |    ⟺
. | . | . |    sender(e) <c sender(e')
```

```
. | . | . |    ∨ sender(e) = sender(e’) & index(e)<index(e’))
. | . | . 18. m : ℤ
. | . | . 19. 0<m
. | . | . 20. ∀l:IdLnk, t:ℕ.
. | . | . |   m-1≤||snds(l;t)|| ⇒ (∃t’:ℕ. t≤t’ & m-1≤||rcvs(l;t’)||)
. | . | . 21. l : IdLnk
. | . | . 22. t : ℕ
. | . | . 23. m≤||snds(l;t)||
. | . | . 24. t’ : ℕ
. | . | . 25. t≤t’
. | . | . 26. m-1≤||rcvs(l;t’)||
. | . | . 27. t’@0 : ℕ
. | . | . 28. t’≤t’@0
. | . | . 29. isrcv(l;a(destination(l);t’@0))
. | . | . 30. ¬t’@0+1 = 0
. | . | . ⊢  m≤||filter(λa.isrcv(l;a);map(λt1.
. | . | . |  a(destination(l);t1);upto(t’@0) @ [t’@0]))||
. | . | . | by (RWO
. | . | . |     Thm* ∀f,as’:Top, A:Type, as:A List.
. | . | . |          map(f;as @ as’) ~ (map(f;as) @ map(f;as’))
. | . | . |     {map_append_sq}
. | . | . |     0)
. | . | . |    THEN
. | . | . |    (RWO
. | . | . |     Thm* ∀P:(T→𝔹), A:T List, B:Top.
. | . | . |          filter(P;A @ B) ~ (filter(P;A) @ filter(P;B))
. | . | . |     {filter_append_sq}
. | . | . |     0)
. | . | . |    THEN
. | . | . |    (Fold ‘w-rcvs‘ 0)
. | . | . |    THEN
. | . | . |    (Reduce 0)
. | . | . |    THEN
. | . | . |    SplitOnConclITE
. | . | . |
. | . | . .....truecase .....
. | . | . 1. the_w : World
. | . | . 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . | . |  ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . | . 3. ∀i:Id, t:ℕ.
. | . | . |  isnull(a(i;t))
. | . | . |  ⇒
. | . | . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . | . 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . | . |  isrcv(l;a(i;t))
. | . | . |  ⇒
. | . | . |  destination(l) = i
. | . | . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . | . 5. ∀l:IdLnk, t:ℕ.
. | . | . |  ∃t’:ℕ.
. | . | . |  t≤t’ & isrcv(l;a(destination(l);t’)) ∨ queue(l;t’) = nil
. | . | . 6. ∀e,e’:E. loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
. | . | . 7. ∀e:E. first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
. | . | . 8. ∀e:E.
. | . | . |  ¬first(e)
```

```
. | . | . |  ⇒
. | . | . |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . | . 9. ∀e:E.
. | . | . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . | . 10. Trans e,e':E. e <c e'
. | . | . 11. SWellFounded(e <c e')
. | . | . 12. ∀e:E.
. | . | . |   isrcv(kind(e))
. | . | . |   ⇒
. | . | . |   sends(lnk(kind(e));sender(e))[index(e)]
. | . | . |   =
. | . | . |   msg(lnk(kind(e));tag(kind(e));val(e))
. | . | . 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . | . 14. ∀e,e':E.
. | . | . |   e <c e'
. | . | . |   ⇒
. | . | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . | . 15. ∀e:E.
. | . | . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . | . 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . | . 17. ∀e,e':E.
. | . | . |   isrcv(kind(e))
. | . | . |   ⇒
. | . | . |   isrcv(kind(e'))
. | . | . |   ⇒
. | . | . |   lnk(kind(e)) = lnk(kind(e'))
. | . | . |   ⇒
. | . | . |   (e <c e'
. | . | . |    ⟺
. | . | . |    sender(e) <c sender(e')
. | . | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . | . 18. m : ℤ
. | . | . 19. 0<m
. | . | . 20. ∀l:IdLnk, t:ℕ.
. | . | . |   m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . | . 21. l : IdLnk
. | . | . 22. t : ℕ
. | . | . 23. m≤||snds(l;t)||
. | . | . 24. t' : ℕ
. | . | . 25. t≤t'
. | . | . 26. m-1≤||rcvs(l;t')||
. | . | . 27. t'@0 : ℕ
. | . | . 28. t'≤t'@0
. | . | . 29. isrcv(l;a(destination(l);t'@0))
. | . | . 30. ¬t'@0+1 = 0
. | . | . 31. isrcv(l;a(destination(l);t'@0))
. | . | . ⊢  m≤||rcvs(l;t'@0) @ [a(destination(l);t'@0)]||
. | . | . | by RWO
. | . | . |    Thm* ∀as,bs:T List. ||as @ bs|| = ||as||+||bs|| ∈ ℤ
. | . | . |    {length_append}
. | . | . |    0
. | . | . |    THEN
. | . | . |    Reduce 0
```

```
. | . | . |
. | . | . 1. the_w : World
. | . | . 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . | . |  ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . | . 3. ∀i:Id, t:ℕ.
. | . | . |  isnull(a(i;t))
. | . | . |  ⇒
. | . | . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . | . 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . | . |  isrcv(l;a(i;t))
. | . | . |  ⇒
. | . | . |  destination(l) = i
. | . | . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . | . 5. ∀l:IdLnk, t:ℕ.
. | . | . |  ∃t':ℕ.
. | . | . |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . | . 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . | . 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . | . 8. ∀e:E.
. | . | . |  ¬first(e)
. | . | . |  ⇒
. | . | . |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . | . 9. ∀e:E.
. | . | . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . | . 10. Trans e,e':E. e <c e'
. | . | . 11. SWellFounded(e <c e')
. | . | . 12. ∀e:E.
. | . | . |   isrcv(kind(e))
. | . | . |   ⇒
. | . | . |   sends(lnk(kind(e));sender(e))[index(e)]
. | . | . |   =
. | . | . |   msg(lnk(kind(e));tag(kind(e));val(e))
. | . | . 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . | . 14. ∀e,e':E.
. | . | . |   e <c e'
. | . | . |   ⇒
. | . | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . | . 15. ∀e:E.
. | . | . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . | . 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . | . 17. ∀e,e':E.
. | . | . |   isrcv(kind(e))
. | . | . |   ⇒
. | . | . |   isrcv(kind(e'))
. | . | . |   ⇒
. | . | . |   lnk(kind(e)) = lnk(kind(e'))
. | . | . |   ⇒
. | . | . |   (e <c e'
. | . | . |    ⟺
. | . | . |    sender(e) <c sender(e')
. | . | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . | . 18. m : ℤ
. | . | . 19. 0<m
```

```
. | . | . 20. ∀l:IdLnk, t:ℕ.
. | . | . |   m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . | . 21. l : IdLnk
. | . | . 22. t : ℕ
. | . | . 23. m≤||snds(l;t)||
. | . | . 24. t' : ℕ
. | . | . 25. t≤t'
. | . | . 26. m-1≤||rcvs(l;t')||
. | . | . 27. t'@0 : ℕ
. | . | . 28. t'≤t'@0
. | . | . 29. isrcv(l;a(destination(l);t'@0))
. | . | . 30. ¬t'@0+1 = 0
. | . | . 31. isrcv(l;a(destination(l);t'@0))
. | . | . ⊢  m≤||rcvs(l;t'@0)||+1
. | . | . | by Assert (||rcvs(l;t')||≤||rcvs(l;t'@0)||)
. | . | . |
. | . | . .....assertion .....
. | . | . 1. the_w : World
. | . | . 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . | . |  ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . | . 3. ∀i:Id, t:ℕ.
. | . | . |  isnull(a(i;t))
. | . | . |  ⇒
. | . | . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . | . 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . | . |  isrcv(l;a(i;t))
. | . | . |  ⇒
. | . | . |  destination(l) = i
. | . | . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . | . 5. ∀l:IdLnk, t:ℕ.
. | . | . |  ∃t':ℕ.
. | . | . |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . | . 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . | . 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . | . 8. ∀e:E.
. | . | . |  ¬first(e)
. | . | . |  ⇒
. | . | . |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . | . 9. ∀e:E.
. | . | . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . | . 10. Trans e,e':E. e <c e'
. | . | . 11. SWellFounded(e <c e')
. | . | . 12. ∀e:E.
. | . | . |   isrcv(kind(e))
. | . | . |   ⇒
. | . | . |   sends(lnk(kind(e));sender(e))[index(e)]
. | . | . |   =
. | . | . |   msg(lnk(kind(e));tag(kind(e));val(e))
. | . | . 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . | . 14. ∀e,e':E.
. | . | . |   e <c e'
. | . | . |   ⇒
. | . | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
```

```
. | . | . 15. ∀e:E.
. | . | . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . | . 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . | . 17. ∀e,e':E.
. | . | . |   isrcv(kind(e))
. | . | . |   ⇒
. | . | . |   isrcv(kind(e'))
. | . | . |   ⇒
. | . | . |   lnk(kind(e)) = lnk(kind(e'))
. | . | . |   ⇒
. | . | . |   (e <c e'
. | . | . |    ⟺
. | . | . |    sender(e) <c sender(e')
. | . | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . | . 18. m : ℤ
. | . | . 19. 0<m
. | . | . 20. ∀l:IdLnk, t:ℕ.
. | . | . |   m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . | . 21. l : IdLnk
. | . | . 22. t : ℕ
. | . | . 23. m≤||snds(l;t)||
. | . | . 24. t' : ℕ
. | . | . 25. t≤t'
. | . | . 26. m-1≤||rcvs(l;t')||
. | . | . 27. t'@0 : ℕ
. | . | . 28. t'≤t'@0
. | . | . 29. isrcv(l;a(destination(l);t'@0))
. | . | . 30. ¬t'@0+1 = 0
. | . | . 31. isrcv(l;a(destination(l);t'@0))
. | . | . ⊢  ||rcvs(l;t')||≤||rcvs(l;t'@0)||
. | . | . | by (BackThru
. | . | . |     Thm* ∀l1,l2:T List. l1 ≤ l2 ⇒ ||l1||≤||l2||
. | . | . |     {iseg_length})
. | . | . |    THEN
. | . | . |    (Unfold `w-rcvs` 0)
. | . | . |
. | . | . 1. the_w : World
. | . | . 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . | .    ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . | . 3. ∀i:Id, t:ℕ.
. | . | .    isnull(a(i;t))
. | . | .    ⇒
. | . | .    (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . | . 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . | .    isrcv(l;a(i;t))
. | . | .    ⇒
. | . | .    destination(l) = i
. | . | .    & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . | . 5. ∀l:IdLnk, t:ℕ.
. | . | .    ∃t':ℕ.
. | . | .    t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . | . 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . | . 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . | . 8. ∀e:E.
. | . | .    ¬first(e)
```

```
. | . | .    ⇒
. | . | .    loc(pred(e)) = loc(e) & pred(e) <c e
. | . | .    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . | . 9. ∀e:E.
. | . | .    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . | . 10. Trans e,e':E. e <c e'
. | . | . 11. SWellFounded(e <c e')
. | . | . 12. ∀e:E.
. | . | .     isrcv(kind(e))
. | . | .     ⇒
. | . | .     sends(lnk(kind(e));sender(e))[index(e)]
. | . | .     =
. | . | .     msg(lnk(kind(e));tag(kind(e));val(e))
. | . | . 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . | . 14. ∀e,e':E.
. | . | .     e <c e'
. | . | .     ⇒
. | . | .     ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . | .     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . | . 15. ∀e:E.
. | . | .     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . | . 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . | . 17. ∀e,e':E.
. | . | .     isrcv(kind(e))
. | . | .     ⇒
. | . | .     isrcv(kind(e'))
. | . | .     ⇒
. | . | .     lnk(kind(e)) = lnk(kind(e'))
. | . | .     ⇒
. | . | .     (e <c e'
. | . | .      ⇐⇒
. | . | .      sender(e) <c sender(e')
. | . | .      ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . | . 18. m : ℤ
. | . | . 19. 0<m
. | . | . 20. ∀l:IdLnk, t:ℕ.
. | . | .     m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . | . 21. l : IdLnk
. | . | . 22. t : ℕ
. | . | . 23. m≤||snds(l;t)||
. | . | . 24. t' : ℕ
. | . | . 25. t≤t'
. | . | . 26. m-1≤||rcvs(l;t')||
. | . | . 27. t'@0 : ℕ
. | . | . 28. t'≤t'@0
. | . | . 29. isrcv(l;a(destination(l);t'@0))
. | . | . 30. ¬t'@0+1 = 0
. | . | . 31. isrcv(l;a(destination(l);t'@0))
. | . | . ⊢  filter(λa.isrcv(l;a);map(λt1.
. | . | .    a(destination(l);t1);upto(t'))) ≤ filter(λa.
. | . | .    isrcv(l;a);map(λt1.a(destination(l);t1);upto(t'@0)))
. | . | .   by (BackThru
. | . | .       Thm* ∀P:(T→𝔹), L2,L1:T List.
. | . | .            L1 ≤ L2 ⇒ filter(P;L1) ≤ filter(P;L2)
. | . | .       {filter_iseg})
```

```
. | . | .       THEN
. | . | .       (BackThru
. | . | .        Thm* ∀f:(A→B), L1,L2:A List.
. | . | .             L1 ≤ L2 ⇒ map(f;L1) ≤ map(f;L2)
. | . | .        {iseg_map})
. | . | .       THEN
. | . | .       (BackThru
. | . | .        Thm* ∀i,j:ℕ. i≤j ⇒ upto(i) ≤ upto(j)    {upto_iseg})
. | . |
. | . .\
. | . . 1. the_w : World
. | . . 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . . |  ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . . 3. ∀i:Id, t:ℕ.
. | . . |  isnull(a(i;t))
. | . . |  ⇒
. | . . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . . 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . . |  isrcv(l;a(i;t))
. | . . |  ⇒
. | . . |  destination(l) = i
. | . . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . . 5. ∀l:IdLnk, t:ℕ.
. | . . |  ∃t':ℕ.
. | . . |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . . 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . . 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . . 8. ∀e:E.
. | . . |  ¬first(e)
. | . . |  ⇒
. | . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . . 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . . 10. Trans e,e':E. e <c e'
. | . . 11. SWellFounded(e <c e')
. | . . 12. ∀e:E.
. | . . |   isrcv(kind(e))
. | . . |   ⇒
. | . . |   sends(lnk(kind(e));sender(e))[index(e)]
. | . . |   =
. | . . |   msg(lnk(kind(e));tag(kind(e));val(e))
. | . . 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . . 14. ∀e,e':E.
. | . . |   e <c e'
. | . . |   ⇒
. | . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . . 15. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . . 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . . 17. ∀e,e':E.
. | . . |   isrcv(kind(e))
. | . . |   ⇒
. | . . |   isrcv(kind(e'))
. | . . |   ⇒
. | . . |   lnk(kind(e)) = lnk(kind(e'))
```

```
. | . . |   ⇒
. | . . |   (e <c e'
. | . . |    ⟺
. | . . |    sender(e) <c sender(e')
. | . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . . 18. m : ℤ
. | . . 19. 0<m
. | . . 20. ∀l:IdLnk, t:ℕ.
. | . . |   m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . . 21. l : IdLnk
. | . . 22. t : ℕ
. | . . 23. m≤||snds(l;t)||

. | . . 24. t' : ℕ
. | . . 25. t≤t'
. | . . 26. m-1≤||rcvs(l;t')||
. | . . 27. t'@0 : ℕ
. | . . 28. t'≤t'@0
. | . . 29. queue(l;t'@0) = nil
. | . . ⊢  ∃t':ℕ. t≤t' & m≤||rcvs(l;t')||
. | . . | by InstConcl [t'@0] THEN Unfold `w-queue` -1
. | . . |    THEN
. | . . |    RWO "length_zero<" -1
. | . . |
. | . . 1. the_w : World
. | . . 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . . |  ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . . 3. ∀i:Id, t:ℕ.
. | . . |  isnull(a(i;t))
. | . . |  ⇒
. | . . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . . 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . . |  isrcv(l;a(i;t))
. | . . |  ⇒
. | . . |  destination(l) = i
. | . . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . . 5. ∀l:IdLnk, t:ℕ.
. | . . |  ∃t':ℕ.
. | . . |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . . 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . . 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . . 8. ∀e:E.
. | . . |  ¬first(e)
. | . . |  ⇒
. | . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . . 9. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . . 10. Trans e,e':E. e <c e'
. | . . 11. SWellFounded(e <c e')
. | . . 12. ∀e:E.
. | . . |   isrcv(kind(e))
. | . . |   ⇒
. | . . |   sends(lnk(kind(e));sender(e))[index(e)]
. | . . |   =
. | . . |   msg(lnk(kind(e));tag(kind(e));val(e))
. | . . 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
```

```
. | . . 14. ∀e,e':E.
. | . . |   e <c e'
. | . . |   ⇒
. | . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . . 15. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . . 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . . 17. ∀e,e':E.
. | . . |   isrcv(kind(e))
. | . . |   ⇒
. | . . |   isrcv(kind(e'))
. | . . |   ⇒
. | . . |   lnk(kind(e)) = lnk(kind(e'))
. | . . |   ⇒
. | . . |   (e <c e'
. | . . |    ⟺
. | . . |    sender(e) <c sender(e')
. | . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . . 18. m : ℤ
. | . . 19. 0<m
. | . . 20. ∀l:IdLnk, t:ℕ.
. | . . |   m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . . 21. l : IdLnk
. | . . 22. t : ℕ
. | . . 23. m≤||snds(l;t)||
. | . . 24. t' : ℕ
. | . . 25. t≤t'
. | . . 26. m-1≤||rcvs(l;t')||
. | . . 27. t'@0 : ℕ
. | . . 28. t'≤t'@0
. | . . 29. ||nth_tl(||rcvs(l;t'@0)||;snds(l;t'@0))|| = 0
. | . . ⊢  m≤||rcvs(l;t'@0)||
. | . . | by RWO
. | . . |    Thm* ∀r:ℕ, L:Top List.
. | . . |         ||nth_tl(r;L)|| = if r<₂||L||→ ||L||-r else 0 fi
. | . . |    {general_length_nth_tl}
. | . . |    -1
. | . . |    THENA
. | . . |    Thin -1
. | . . |    THEN
. | . . |    Assert (||snds(l;t'@0)||≤||rcvs(l;t'@0)||)
. | . . |
. | . . |\
. | . . | .....assertion .....
. | . . | 1. the_w : World
. | . . | 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . . |    ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . . | 3. ∀i:Id, t:ℕ.
. | . . |    isnull(a(i;t))
. | . . |    ⇒
. | . . |    (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . . | 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . . |    isrcv(l;a(i;t))
. | . . |    ⇒
. | . . |    destination(l) = i
```

```
. | . . |    & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . . | 5. ∀l:IdLnk, t:ℕ.
. | . . |    ∃t':ℕ.
. | . . |    t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . . | 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . . | 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . . | 8. ∀e:E.
. | . . |    ¬first(e)
. | . . |    ⇒
. | . . |    loc(pred(e)) = loc(e) & pred(e) <c e
. | . . |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . . | 9. ∀e:E.
. | . . |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . . | 10. Trans e,e':E. e <c e'
. | . . | 11. SWellFounded(e <c e')
. | . . | 12. ∀e:E.
. | . . |     isrcv(kind(e))
. | . . |     ⇒
. | . . |     sends(lnk(kind(e));sender(e))[index(e)]
. | . . |     =
. | . . |     msg(lnk(kind(e));tag(kind(e));val(e))
. | . . | 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . . | 14. ∀e,e':E.
. | . . |     e <c e'
. | . . |     ⇒
. | . . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . . |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . . | 15. ∀e:E.
. | . . |     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . . | 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . . | 17. ∀e,e':E.
. | . . |     isrcv(kind(e))
. | . . |     ⇒
. | . . |     isrcv(kind(e'))
. | . . |     ⇒
. | . . |     lnk(kind(e)) = lnk(kind(e'))
. | . . |     ⇒
. | . . |     (e <c e'
. | . . |      ⟺
. | . . |      sender(e) <c sender(e')
. | . . |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . . | 18. m : ℤ
. | . . | 19. 0<m
. | . . | 20. ∀l:IdLnk, t:ℕ.
. | . . |     m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . . | 21. l : IdLnk
. | . . | 22. t : ℕ
. | . . | 23. m≤||snds(l;t)||
. | . . | 24. t' : ℕ
. | . . | 25. t≤t'
. | . . | 26. m-1≤||rcvs(l;t')||
. | . . | 27. t'@0 : ℕ
. | . . | 28. t'≤t'@0
. | . . | 29. if ||rcvs(l;t'@0)||<2||snds(l;t'@0)||→
. | . . |        ||snds(l;t'@0)||-||rcvs(l;t'@0)||
```

```
. | . . |     else 0 fi
. | . . |     =
. | . . |     0
. | . . | ⊢  ||snds(l;t'@0)||≤||rcvs(l;t'@0)||
. | . . |   by MoveToConcl -1 THEN SplitOnConclITE
. | . . |
. | . . .\
. | . . . 1. the_w : World
. | . . . 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . . . |  ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . . . 3. ∀i:Id, t:ℕ.
. | . . . |  isnull(a(i;t))
. | . . . |  ⇒
. | . . . |  (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . . . 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . . . |  isrcv(l;a(i;t))
. | . . . |  ⇒
. | . . . |  destination(l) = i
. | . . . |  & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . . . 5. ∀l:IdLnk, t:ℕ.
. | . . . |  ∃t':ℕ.
. | . . . |  t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . . . 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . . . 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . . . 8. ∀e:E.
. | . . . |  ¬first(e)
. | . . . |  ⇒
. | . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. | . . . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . . . 9. ∀e:E.
. | . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . . . 10. Trans e,e':E. e <c e'
. | . . . 11. SWellFounded(e <c e')
. | . . . 12. ∀e:E.
. | . . . |   isrcv(kind(e))
. | . . . |   ⇒
. | . . . |   sends(lnk(kind(e));sender(e))[index(e)]
. | . . . |   =
. | . . . |   msg(lnk(kind(e));tag(kind(e));val(e))
. | . . . 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . . . 14. ∀e,e':E.
. | . . . |   e <c e'
. | . . . |   ⇒
. | . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . . . 15. ∀e:E.
. | . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . . . 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . . . 17. ∀e,e':E.
. | . . . |   isrcv(kind(e))
. | . . . |   ⇒
. | . . . |   isrcv(kind(e'))
. | . . . |   ⇒
. | . . . |   lnk(kind(e)) = lnk(kind(e'))
. | . . . |   ⇒
```

```
. | . . . |   (e <c e'
. | . . . |    ⟺
. | . . . |    sender(e) <c sender(e')
. | . . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . . . 18. m : ℤ
. | . . . 19. 0<m
. | . . . 20. ∀l:IdLnk, t:ℕ.
. | . . . |   m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . . . 21. l : IdLnk
. | . . . 22. t : ℕ
. | . . . 23. m≤||snds(l;t)||
. | . . . 24. t' : ℕ
. | . . . 25. t≤t'
. | . . . 26. m-1≤||rcvs(l;t')||
. | . . . 27. t'@0 : ℕ
. | . . . 28. t'≤t'@0
. | . . . 29. if ||rcvs(l;t'@0)||<_2||snds(l;t'@0)||→
. | . . . |      ||snds(l;t'@0)||-||rcvs(l;t'@0)||
. | . . . |   else 0 fi
. | . . . |   =
. | . . . |   0
. | . . . 30. ||snds(l;t'@0)||≤||rcvs(l;t'@0)||
. | . . . ⊢  m≤||rcvs(l;t'@0)||
. | . . . | by Assert (||snds(l;t)||≤||snds(l;t'@0)||)
. | . . . |
. | . . . .....assertion .....
. | . . . 1. the_w : World
. | . . . 2. ∀i:Id, t:ℕ, l:IdLnk.
. | . . .    ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. | . . . 3. ∀i:Id, t:ℕ.
. | . . .    isnull(a(i;t))
. | . . .    ⇒
. | . . .    (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil
. | . . . 4. ∀i:Id, t:ℕ, l:IdLnk.
. | . . .    isrcv(l;a(i;t))
. | . . .    ⇒
. | . . .    destination(l) = i
. | . . .    & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t))
. | . . . 5. ∀l:IdLnk, t:ℕ.
. | . . .    ∃t':ℕ.
. | . . .    t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil
. | . . . 6. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. | . . . 7. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. | . . . 8. ∀e:E.
. | . . .    ¬first(e)
. | . . .    ⇒
. | . . .    loc(pred(e)) = loc(e) & pred(e) <c e
. | . . .    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. | . . . 9. ∀e:E.
. | . . .    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. | . . . 10. Trans e,e':E. e <c e'
. | . . . 11. SWellFounded(e <c e')
. | . . . 12. ∀e:E.
. | . . .     isrcv(kind(e))
. | . . .     ⇒
```

```
. | . . .     sends(lnk(kind(e));sender(e))[index(e)]
. | . . .     =
. | . . .     msg(lnk(kind(e));tag(kind(e));val(e))
. | . . . 13. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. | . . . 14. ∀e,e':E.
. | . . .     e <c e'
. | . . .     ⇒
. | . . .     ¬first(e') & e <c pred(e') ∨ e = pred(e')
. | . . .     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. | . . . 15. ∀e:E.
. | . . .     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. | . . . 16. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. | . . . 17. ∀e,e':E.
. | . . .     isrcv(kind(e))
. | . . .     ⇒
. | . . .     isrcv(kind(e'))
. | . . .     ⇒
. | . . .     lnk(kind(e)) = lnk(kind(e'))
. | . . .     ⇒
. | . . .     (e <c e'
. | . . .      ⟺
. | . . .      sender(e) <c sender(e')
. | . . .      ∨ sender(e) = sender(e') & index(e)<index(e'))
. | . . . 18. m : ℤ
. | . . . 19. 0<m
. | . . . 20. ∀l:IdLnk, t:ℕ.
. | . . .     m-1≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m-1≤||rcvs(l;t')||)
. | . . . 21. l : IdLnk
. | . . . 22. t : ℕ
. | . . . 23. m≤||snds(l;t)||
. | . . . 24. t' : ℕ
. | . . . 25. t≤t'
. | . . . 26. m-1≤||rcvs(l;t')||
. | . . . 27. t'@0 : ℕ
. | . . . 28. t'≤t'@0
. | . . . 29. if ||rcvs(l;t'@0)||<₂||snds(l;t'@0)||→
. | . . .        ||snds(l;t'@0)||-||rcvs(l;t'@0)||
. | . . .     else 0 fi
. | . . .     =
. | . . .     0
. | . . . 30. ||snds(l;t'@0)||≤||rcvs(l;t'@0)||
. | . . . ⊢  ||snds(l;t)||≤||snds(l;t'@0)||
. | . . .   by (BackThru
. | . . .       Thm* ∀l1,l2:T List. l1 ≤ l2 ⇒ ||l1||≤||l2||
. | . . .       {iseg_length})
. | . . .      THEN
. | . . .      (Unfold `w-snds` 0)
. | . . .      THEN
. | . . .      (BackThru
. | . . .       Thm* ∀ll1,ll2:(T List) List.
. | . . .            ll1 ≤ ll2 ⇒ concat(ll1) ≤ concat(ll2)
. | . . .       {concat_iseg})
. | . . .      THEN
. | . . .      (BackThru
. | . . .       Thm* ∀f:(A→B), L1,L2:A List.
```

```
. | . . .           L1 ≤ L2 ⇒ map(f;L1) ≤ map(f;L2)
. | . . .       {iseg_map})
. | . . .      THEN
. | . . .      (BackThru
. | . . .       Thm* ∀i,j:ℕ. i≤j ⇒ upto(i) ≤ upto(j)    {upto_iseg})
. |
. .\
. . 1. the_w : World
. . 2. FairFifo
. . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . 5. ∀e:E.
. . |  ¬first(e)
. . |  ⇒
. . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . 7. Trans e,e':E. e <c e'
. . 8. SWellFounded(e <c e')
. . 9. ∀e:E.
. . |  isrcv(kind(e))
. . |  ⇒
. . |  sends(lnk(kind(e));sender(e))[index(e)]
. . |  =
. . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . 11. ∀e,e':E.
. . |   e <c e'
. . |   ⇒
. . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . 14. ∀e,e':E.
. . |   isrcv(kind(e))
. . |   ⇒
. . |   isrcv(kind(e'))
. . |   ⇒
. . |   lnk(kind(e)) = lnk(kind(e'))
. . |   ⇒
. . |   (e <c e'
. . |    ⟺
. . |    sender(e) <c sender(e')
. . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . ⊢  ∀e:E, l:IdLnk, n:ℕ||sends(l;e)||.
. . |  ∃e':E.
. . |  isrcv(kind(e'))
. . |  & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. . | by Auto THEN InstHyp [||snds(l;time(e))||+n+1;l;time(e)+1] -4
. . |
. . |\
. . | .....antecedent .....
. . | 1. the_w : World
```

```
. . | 2. FairFifo
. . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . | 5. ∀e:E.
. . | |  ¬first(e)
. . | |  ⇒
. . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . | 7. Trans e,e':E. e <c e'
. . | 8. SWellFounded(e <c e')
. . | 9. ∀e:E.
. . | |  isrcv(kind(e))
. . | |  ⇒
. . | |  sends(lnk(kind(e));sender(e))[index(e)]
. . | |  =
. . | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . | 11. ∀e,e':E.
. . | |   e <c e'
. . | |   ⇒
. . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . | 14. ∀e,e':E.
. . | |   isrcv(kind(e))
. . | |   ⇒
. . | |   isrcv(kind(e'))
. . | |   ⇒
. . | |   lnk(kind(e)) = lnk(kind(e'))
. . | |   ⇒
. . | |   (e <c e'
. . | |    ⟺
. . | |    sender(e) <c sender(e')
. . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . | |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . | 16. e : E
. . | 17. l : IdLnk
. . | 18. n : ℕ||sends(l;e)||
. . | ⊢  ||snds(l;time(e))||+n+1≤||snds(l;time(e)+1)||
. . | | by Unfold `w-snds` 0 THEN RW (AddrC [2] (RecUnfoldC `upto`)) 0
. . | |    THEN
. . | |    SplitOnConclITE
. . | |    THEN
. . | |    Try (Assert False THEN Complete Auto)
. . | |    THEN
. . | |    Subst' (time(e)+1-1 = time(e)) 0
. . | |    THEN
. . | |    RWO
. . | |    Thm* ∀f,as':Top, A:Type, as:A List.
. . | |         map(f;as @ as') ~ (map(f;as) @ map(f;as'))
. . | |    {map_append_sq}
. . | |    0
```

```
. . | |    THEN
. . | |    Fold `w-snds` 0
. . | |    THEN
. . | |    Reduce 0
. . | |
. . | 1. the_w : World
. . | 2. FairFifo
. . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . | 5. ∀e:E.
. . | |  ¬first(e)
. . | |  ⇒
. . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . | 7. Trans e,e':E. e <c e'
. . | 8. SWellFounded(e <c e')
. . | 9. ∀e:E.
. . | |  isrcv(kind(e))
. . | |  ⇒
. . | |  sends(lnk(kind(e));sender(e))[index(e)]
. . | |  =
. . | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . | 11. ∀e,e':E.
. . | |   e <c e'
. . | |   ⇒
. . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . | 14. ∀e,e':E.
. . | |   isrcv(kind(e))
. . | |   ⇒
. . | |   isrcv(kind(e'))
. . | |   ⇒
. . | |   lnk(kind(e)) = lnk(kind(e'))
. . | |   ⇒
. . | |   (e <c e'
. . | |    ⟺
. . | |    sender(e) <c sender(e')
. . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . | |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . | 16. e : E
. . | 17. l : IdLnk
. . | 18. n : ℕ||sends(l;e)||
. . | 19. ¬time(e)+1 = 0
. . | ⊢  ||snds(l;time(e))||+n+1≤||concat(map(λt1.
. . | |  m(l;t1);upto(time(e)))
. . | |  @ [m(l;time(e))])||
. . | | by (RWO
. . | |     Thm* ∀ll1,ll2:(Top List) List.
. . | |          concat(ll1 @ ll2) ~ (concat(ll1) @ concat(ll2))
. . | |     {concat_append}
```

```
. . | |     0)
. . | |    THEN
. . | |    (RWO
. . | |     Thm* ∀l:Top List, ll:(Top List) List.
. . | |          concat([l / ll]) ~ (l @ concat(ll))
. . | |     {concat_DASH_cons}
. . | |     0)
. . | |    THEN
. . | |    (Reduce 0)
. . | |    THEN
. . | |    (RWO Thm* ∀l:T List. (l @ nil) ~ l    {append_nil_sq} 0)
. . | |    THEN
. . | |    (Fold `w-snds` 0)
. . | |
. . | 1. the_w : World
. . | 2. FairFifo
. . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . | 5. ∀e:E.
. . | |  ¬first(e)
. . | |  ⇒
. . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . | 7. Trans e,e':E. e <c e'
. . | 8. SWellFounded(e <c e')
. . | 9. ∀e:E.
. . | |  isrcv(kind(e))
. . | |  ⇒
. . | |  sends(lnk(kind(e));sender(e))[index(e)]
. . | |  =
. . | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . | 11. ∀e,e':E.
. . | |   e <c e'
. . | |   ⇒
. . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . | 14. ∀e,e':E.
. . | |   isrcv(kind(e))
. . | |   ⇒
. . | |   isrcv(kind(e'))
. . | |   ⇒
. . | |   lnk(kind(e)) = lnk(kind(e'))
. . | |   ⇒
. . | |   (e <c e'
. . | |    ⟺
. . | |    sender(e) <c sender(e')
. . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . | |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . | 16. e : E
. . | 17. l : IdLnk
```

```
. . | 18. n : ℕ||sends(l;e)||
. . | 19. ¬time(e)+1 = 0
. . | ⊢  ||snds(l;time(e))||+n+1≤||snds(l;time(e)) @ m(l;time(e))||
. . | | by RWO
. . | |    Thm* ∀as,bs:T List. ||as @ bs|| = ||as||+||bs|| ∈ ℤ
. . | |    {length_append}
. . | |    0
. . | |    THEN
. . | |    Unfold `w-sends` -2
. . | |    THEN
. . | |    Unfold `w-ml` 0
. . | |    THEN
. . | |    Subst'
. . | |    (||onlnk(l;m(source(l);time(e)))||
. . | |     =
. . | |     ||onlnk(l;m(loc(e);time(e)))||)
. . | |    0
. . | |    THEN
. . | |    Try
. . | |    (BackThru
. . | |     Thm* ∀w:World, t:ℕ, l:IdLnk, i:Id. ||onlnk(l;m(i;t))|| ∈ ℤ
. . | |     {w_DASH_onlnk_DASH_m})
. . | |
. . | 1. the_w : World
. . | 2. FairFifo
. . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . | 5. ∀e:E.
. . | |  ¬first(e)
. . | |  ⇒
. . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . | 7. Trans e,e':E. e <c e'
. . | 8. SWellFounded(e <c e')
. . | 9. ∀e:E.
. . | |  isrcv(kind(e))
. . | |  ⇒
. . | |  sends(lnk(kind(e));sender(e))[index(e)]
. . | |  =
. . | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . | 11. ∀e,e':E.
. . | |   e <c e'
. . | |   ⇒
. . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . | 14. ∀e,e':E.
. . | |   isrcv(kind(e))
. . | |   ⇒
. . | |   isrcv(kind(e'))
. . | |   ⇒
. . | |   lnk(kind(e)) = lnk(kind(e'))
```

```
. . | |    ⇒
. . | |    (e <c e'
. . | |     ⟺
. . | |     sender(e) <c sender(e')
. . | |     ∨ sender(e) = sender(e') & index(e)<index(e'))
. . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . | |    m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . | 16. e : E
. . | 17. l : IdLnk
. . | 18. ℕ||onlnk(l;m(loc(e);time(e)))||
. . | 19. ¬time(e)+1 = 0
. . | ⊢  ||onlnk(l;m(source(l);time(e)))||
. . | |  =
. . | |  ||onlnk(l;m(loc(e);time(e)))||
. . | | by Subst (source(l) = loc(e)) 0 THEN Try (Fold `member` 0)
. . | |     THEN
. . | |     Try
. . | |     (BackThru
. . | |      Thm* ∀w:World, t:ℕ, l:IdLnk, i:Id. ||onlnk(l;m(i;t))|| ∈ ℤ
. . | |      {w_DASH_onlnk_DASH_m})
. . | |
. . | .....equality .....
. . | 1. the_w : World
. . | 2. FairFifo
. . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . | 5. ∀e:E.
. . | |  ¬first(e)
. . | |  ⇒
. . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . | 7. Trans e,e':E. e <c e'
. . | 8. SWellFounded(e <c e')
. . | 9. ∀e:E.
. . | |  isrcv(kind(e))
. . | |  ⇒
. . | |  sends(lnk(kind(e));sender(e))[index(e)]
. . | |  =
. . | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . | 11. ∀e,e':E.
. . | |    e <c e'
. . | |    ⇒
. . | |    ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . | |    ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . | 14. ∀e,e':E.
. . | |    isrcv(kind(e))
. . | |    ⇒
. . | |    isrcv(kind(e'))
. . | |    ⇒
. . | |    lnk(kind(e)) = lnk(kind(e'))
. . | |    ⇒
```

```
. . | |   (e <c e'
. . | |    ⟺
. . | |    sender(e) <c sender(e')
. . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . | |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . | 16. e : E
. . | 17. l : IdLnk
. . | 18. ℕ||onlnk(l;m(loc(e);time(e)))||
. . | 19. ¬time(e)+1 = 0
. . | ⊢  source(l) = loc(e)
. . | | by (Decide source(l) = loc(e)) THEN (RW assert_pushdownC -1)
. . | |
. . | 1. the_w : World
. . | 2. FairFifo
. . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . | 5. ∀e:E.
. . | |  ¬first(e)
. . | |  ⇒
. . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . | 7. Trans e,e':E. e <c e'
. . | 8. SWellFounded(e <c e')
. . | 9. ∀e:E.
. . | |  isrcv(kind(e))
. . | |  ⇒
. . | |  sends(lnk(kind(e));sender(e))[index(e)]
. . | |  =
. . | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . | 11. ∀e,e':E.
. . | |   e <c e'
. . | |   ⇒
. . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . | 14. ∀e,e':E.
. . | |   isrcv(kind(e))
. . | |   ⇒
. . | |   isrcv(kind(e'))
. . | |   ⇒
. . | |   lnk(kind(e)) = lnk(kind(e'))
. . | |   ⇒
. . | |   (e <c e'
. . | |    ⟺
. . | |    sender(e) <c sender(e')
. . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . | |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . | 16. e : E
. . | 17. l : IdLnk
. . | 18. ℕ||onlnk(l;m(loc(e);time(e)))||
```

```
. . | 19. ¬time(e)+1 = 0
. . | 20. ¬source(l) = loc(e)
. . | ⊢  source(l) = loc(e)
. . | | by Analyze 2 THEN InstHyp [loc(e);time(e);l] 2
. . | |
. . | 1. the_w : World
. . | 2. ∀i:Id, t:ℕ, l:IdLnk. ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. . | 3. (∀i:Id, t:ℕ.
. . |     isnull(a(i;t))
. . |     ⇒
. . |     (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil)
. . |    & (∀i:Id, t:ℕ, l:IdLnk.
. . |       isrcv(l;a(i;t))
. . |       ⇒
. . |       destination(l) = i
. . |       & ||queue(l;t)||≥1 & hd(queue(l;t)) = msg(a(i;t)))
. . |    & (∀l:IdLnk, t:ℕ.
. . |       ∃t':ℕ.
. . |       t≤t' & isrcv(l;a(destination(l);t')) ∨ queue(l;t') = nil)
. . | 4. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . | 5. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . | 6. ∀e:E.
. . |    ¬first(e)
. . |    ⇒
. . |    loc(pred(e)) = loc(e) & pred(e) <c e
. . |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . | 7. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . | 8. Trans e,e':E. e <c e'
. . | 9. SWellFounded(e <c e')
. . | 10. ∀e:E.
. . |     isrcv(kind(e))
. . |     ⇒
. . |     sends(lnk(kind(e));sender(e))[index(e)]
. . |     =
. . |     msg(lnk(kind(e));tag(kind(e));val(e))
. . | 11. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . | 12. ∀e,e':E.
. . |     e <c e'
. . |     ⇒
. . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . | 13. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . | 14. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . | 15. ∀e,e':E.
. . |     isrcv(kind(e))
. . |     ⇒
. . |     isrcv(kind(e'))
. . |     ⇒
. . |     lnk(kind(e)) = lnk(kind(e'))
. . |     ⇒
. . |     (e <c e'
. . |      ⟺
. . |      sender(e) <c sender(e')
. . |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. . | 16. ∀m:ℕ, l:IdLnk, t:ℕ.
```

```
. . |     m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . | 17. e : E
. . | 18. l : IdLnk
. . | 19. ℕ||onlnk(l;m(loc(e);time(e)))||
. . | 20. ¬time(e)+1 = 0
. . | 21. ¬source(l) = loc(e)
. . | 22. onlnk(l;m(loc(e);time(e))) = nil
. . | ⊢  source(l) = loc(e)  by HypSubst -1 -4 THEN Reduce -4
. . |
. . .\
. . . 1. the_w : World
. . . 2. FairFifo
. . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . 5. ∀e:E.
. . . |  ¬first(e)
. . . |  ⇒
. . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . 7. Trans e,e':E. e <c e'
. . . 8. SWellFounded(e <c e')
. . . 9. ∀e:E.
. . . |  isrcv(kind(e))
. . . |  ⇒
. . . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . |  =
. . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . 11. ∀e,e':E.
. . . |   e <c e'
. . . |   ⇒
. . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . . 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . 14. ∀e,e':E.
. . . |   isrcv(kind(e))
. . . |   ⇒
. . . |   isrcv(kind(e'))
. . . |   ⇒
. . . |   lnk(kind(e)) = lnk(kind(e'))
. . . |   ⇒
. . . |   (e <c e'
. . . |    ⟺
. . . |    sender(e) <c sender(e')
. . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . 16. e : E
. . . 17. l : IdLnk
. . . 18. n : ℕ||sends(l;e)||
. . . 19. ∃t':ℕ. time(e)+1≤t' & ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . ⊢  ∃e':E.
. . . |  isrcv(kind(e'))
```

```
. . . |  & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. . . | by ExRepD
. . . |    THEN
. . . |    Assert
. . . |    (∃t:ℕ.
. . . |     isrcv(l;a(destination(l);t))
. . . |     & ||rcvs(l;t)|| = ||snds(l;time(e))||+n)
. . . |
. . . |\
. . . | .....assertion .....
. . . | 1. the_w : World
. . . | 2. FairFifo
. . . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . | 5. ∀e:E.
. . . | |  ¬first(e)
. . . | |  ⇒
. . . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . | |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . | 7. Trans e,e':E. e <c e'
. . . | 8. SWellFounded(e <c e')
. . . | 9. ∀e:E.
. . . | |  isrcv(kind(e))
. . . | |  ⇒
. . . | |  sends(lnk(kind(e));sender(e))[index(e)]
. . . | |  =
. . . | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . | 11. ∀e,e':E.
. . . | |   e <c e'
. . . | |   ⇒
. . . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . | |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . | 14. ∀e,e':E.
. . . | |   isrcv(kind(e))
. . . | |   ⇒
. . . | |   isrcv(kind(e'))
. . . | |   ⇒
. . . | |   lnk(kind(e)) = lnk(kind(e'))
. . . | |   ⇒
. . . | |   (e <c e'
. . . | |    ⟺
. . . | |    sender(e) <c sender(e')
. . . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . | |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . | 16. e : E
. . . | 17. l : IdLnk
. . . | 18. n : ℕ||sends(l;e)||
. . . | 19. t' : ℕ
. . . | 20. time(e)+1≤t'
. . . | 21. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
```

```
. . . | ⊢  ∃t:ℕ.
. . . | |  isrcv(l;a(destination(l);t))
. . . | |  & ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . | | by MoveToConcl -1 THEN GenConcl (||snds(l;time(e))||+n = m)
. . . | |    THEN
. . . | |    Thin -1
. . . | |    THEN
. . . | |    Unfold ‘w-rcvs‘ 0
. . . | |    THEN
. . . | |    GenConcl ((λt1.a(destination(l);t1)) = f)
. . . | |    THEN
. . . | |    GenConcl ((λa.isrcv(l;a)) = P)
. . . | |
. . . | 1. the_w : World
. . . | 2. FairFifo
. . . | 3. ∀e,e’:E. loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
. . . | 4. ∀e:E. first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
. . . | 5. ∀e:E.
. . . | |  ¬first(e)
. . . | |  ⇒
. . . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . | |  & (∀e’:E. loc(e’) = loc(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
. . . | 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . | 7. Trans e,e’:E. e <c e’
. . . | 8. SWellFounded(e <c e’)
. . . | 9. ∀e:E.
. . . | |  isrcv(kind(e))
. . . | |  ⇒
. . . | |  sends(lnk(kind(e));sender(e))[index(e)]
. . . | |  =
. . . | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . | 11. ∀e,e’:E.
. . . | |   e <c e’
. . . | |   ⇒
. . . | |   ¬first(e’) & e <c pred(e’) ∨ e = pred(e’)
. . . | |   ∨ isrcv(kind(e’)) & e <c sender(e’) ∨ e = sender(e’)
. . . | 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . | 14. ∀e,e’:E.
. . . | |   isrcv(kind(e))
. . . | |   ⇒
. . . | |   isrcv(kind(e’))
. . . | |   ⇒
. . . | |   lnk(kind(e)) = lnk(kind(e’))
. . . | |   ⇒
. . . | |   (e <c e’
. . . | |    ⟺
. . . | |    sender(e) <c sender(e’)
. . . | |    ∨ sender(e) = sender(e’) & index(e)<index(e’))
. . . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . | |   m≤||snds(l;t)|| ⇒ (∃t’:ℕ. t≤t’ & m≤||rcvs(l;t’)||)
. . . | 16. e : E
. . . | 17. l : IdLnk
. . . | 18. ℕ||sends(l;e)||
```

```
. . . | 19. t' : ℕ
. . . | 20. time(e)+1≤t'
. . . | 21. m : ℕ
. . . | 22. f : ℕ→Action(destination(l))
. . . | 23. (λt1.a(destination(l);t1)) = f
. . . | 24. P : Action(destination(l))→𝔹
. . . | 25. (λa.isrcv(l;a)) = P
. . . | ⊢  m+1≤||filter(P;map(f;upto(t')))||
. . . | |  ⇒
. . . | |  (∃t:ℕ.
. . . | |   isrcv(l;a(destination(l);t))
. . . | |   & ||filter(P;map(f;upto(t)))|| = m)
. . . | | by Assert
. . . | |    (m+1≤||filter(P;map(f;upto(t')))||
. . . | |     ⇒
. . . | |     (∃t:ℕ. P(f(t)) & ||filter(P;map(f;upto(t)))|| = m))
. . . | |
. . . | |\
. . . | | .....assertion .....
. . . | | 1. the_w : World
. . . | | 2. FairFifo
. . . | | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . | | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . | | 5. ∀e:E.
. . . | |    ¬first(e)
. . . | |    ⇒
. . . | |    loc(pred(e)) = loc(e) & pred(e) <c e
. . . | |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . | | 6. ∀e:E.
. . . | |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . | | 7. Trans e,e':E. e <c e'
. . . | | 8. SWellFounded(e <c e')
. . . | | 9. ∀e:E.
. . . | |    isrcv(kind(e))
. . . | |    ⇒
. . . | |    sends(lnk(kind(e));sender(e))[index(e)]
. . . | |    =
. . . | |    msg(lnk(kind(e));tag(kind(e));val(e))
. . . | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . | | 11. ∀e,e':E.
. . . | |     e <c e'
. . . | |     ⇒
. . . | |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . | |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . . | | 12. ∀e:E.
. . . | |     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . | | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . | | 14. ∀e,e':E.
. . . | |     isrcv(kind(e))
. . . | |     ⇒
. . . | |     isrcv(kind(e'))
. . . | |     ⇒
. . . | |     lnk(kind(e)) = lnk(kind(e'))
. . . | |     ⇒
. . . | |     (e <c e'
```

```
. . . | |      ⟺
. . . | |      sender(e) <c sender(e')
. . . | |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . | | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . | |     m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . | | 16. e : E
. . . | | 17. l : IdLnk
. . . | | 18. ℕ||sends(l;e)||
. . . | | 19. t' : ℕ
. . . | | 20. time(e)+1≤t'
. . . | | 21. m : ℕ
. . . | | 22. f : ℕ→Action(destination(l))
. . . | | 23. (λt1.a(destination(l);t1)) = f
. . . | | 24. P : Action(destination(l))→𝔹
. . . | | 25. (λa.isrcv(l;a)) = P
. . . | | ⊢  m+1≤||filter(P;map(f;upto(t')))||
. . . | |    ⇒
. . . | |    (∃t:ℕ. P(f(t)) & ||filter(P;map(f;upto(t)))|| = m)
. . . | |   by Analyze 0
. . . | |      THEN
. . . | |      Using [`t\'',t']
. . . | |      (BackThru
. . . | |       Thm* ∀t',m:ℕ, T:Type, f:(ℕ→T), P:(T→𝔹).
. . . | |            m+1≤||filter(P;map(f;upto(t')))||
. . . | |            ⇒
. . . | |            (∃t:ℕ.
. . . | |             P(f(t)) & ||filter(P;map(f;upto(t)))|| = m ∈ ℤ)
. . . | |       {filter_map_upto2})
. . . | |
. . . | .\
. . . | . 1. the_w : World
. . . | . 2. FairFifo
. . . | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . | . 5. ∀e:E.
. . . | . |  ¬first(e)
. . . | . |  ⇒
. . . | . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . | . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . | . 6. ∀e:E.
. . . | . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . | . 7. Trans e,e':E. e <c e'
. . . | . 8. SWellFounded(e <c e')
. . . | . 9. ∀e:E.
. . . | . |  isrcv(kind(e))
. . . | . |  ⇒
. . . | . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . | . |  =
. . . | . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . | . 11. ∀e,e':E.
. . . | . |   e <c e'
. . . | . |   ⇒
. . . | . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . | . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
```

```
. . . | . 12. ∀e:E.
. . . | . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . | . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . | . 14. ∀e,e':E.
. . . | . |   isrcv(kind(e))
. . . | . |   ⇒
. . . | . |   isrcv(kind(e'))
. . . | . |   ⇒
. . . | . |   lnk(kind(e)) = lnk(kind(e'))
. . . | . |   ⇒
. . . | . |   (e <c e'
. . . | . |    ⟺
. . . | . |    sender(e) <c sender(e')
. . . | . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . | . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . | . |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . | . 16. e : E
. . . | . 17. l : IdLnk
. . . | . 18. ℕ||sends(l;e)||
. . . | . 19. t' : ℕ
. . . | . 20. time(e)+1≤t'
. . . | . 21. m : ℕ
. . . | . 22. f : ℕ→Action(destination(l))
. . . | . 23. (λt1.a(destination(l);t1)) = f
. . . | . 24. P : Action(destination(l))→𝔹
. . . | . 25. (λa.isrcv(l;a)) = P
. . . | . 26. m+1≤||filter(P;map(f;upto(t')))||
. . . | . |   ⇒
. . . | . |   (∃t:ℕ. P(f(t)) & ||filter(P;map(f;upto(t)))|| = m)
. . . | . ⊢  m+1≤||filter(P;map(f;upto(t')))||
. . . | . |  ⇒
. . . | . |  (∃t:ℕ.
. . . | . |   isrcv(l;a(destination(l);t))
. . . | . |   & ||filter(P;map(f;upto(t)))|| = m)
. . . | . | by RepeatFor 3 (ParallelOp -1)
. . . | . |
. . . | . 1. the_w : World
. . . | . 2. FairFifo
. . . | . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . | . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . | . 5. ∀e:E.
. . . | .    ¬first(e)
. . . | .    ⇒
. . . | .    loc(pred(e)) = loc(e) & pred(e) <c e
. . . | .    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . | . 6. ∀e:E.
. . . | .    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . | . 7. Trans e,e':E. e <c e'
. . . | . 8. SWellFounded(e <c e')
. . . | . 9. ∀e:E.
. . . | .    isrcv(kind(e))
. . . | .    ⇒
. . . | .    sends(lnk(kind(e));sender(e))[index(e)]
. . . | .    =
. . . | .    msg(lnk(kind(e));tag(kind(e));val(e))
```

```
. . . | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . | . 11. ∀e,e':E.
. . . | .     e <c e'
. . . | .     ⇒
. . . | .     ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . | .     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . . | . 12. ∀e:E.
. . . | .     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . | . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . | . 14. ∀e,e':E.
. . . | .     isrcv(kind(e))
. . . | .     ⇒
. . . | .     isrcv(kind(e'))
. . . | .     ⇒
. . . | .     lnk(kind(e)) = lnk(kind(e'))
. . . | .     ⇒
. . . | .     (e <c e'
. . . | .      ⟺
. . . | .      sender(e) <c sender(e')
. . . | .      ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . | . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . | .     m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . | . 16. e : E
. . . | . 17. l : IdLnk
. . . | . 18. ℕ||sends(l;e)||
. . . | . 19. t' : ℕ
. . . | . 20. time(e)+1≤t'
. . . | . 21. m : ℕ
. . . | . 22. f : ℕ→Action(destination(l))
. . . | . 23. (λt1.a(destination(l);t1)) = f
. . . | . 24. P : Action(destination(l))→𝔹
. . . | . 25. (λa.isrcv(l;a)) = P
. . . | . 26. m+1≤||filter(P;map(f;upto(t')))||
. . . | . 27. t : ℕ
. . . | . 28. P(f(t))
. . . | . ⊢  isrcv(l;a(destination(l);t))
. . . | .   by ((RevHypSubst -4 -1) THEN (RevHypSubst -6 -1))
. . . | .      THEN
. . . | .      All Reduce
. . . |
. . . .\
. . . . 1. the_w : World
. . . . 2. FairFifo
. . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . 5. ∀e:E.
. . . . |  ¬first(e)
. . . . |  ⇒
. . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . . 7. Trans e,e':E. e <c e'
. . . . 8. SWellFounded(e <c e')
. . . . 9. ∀e:E.
. . . . |  isrcv(kind(e))
```

```
. . . . |  ⇒
. . . . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . |  =
. . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . 11. ∀e,e':E.
. . . . |   e <c e'
. . . . |   ⇒
. . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . . . 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . 14. ∀e,e':E.
. . . . |   isrcv(kind(e))
. . . . |   ⇒
. . . . |   isrcv(kind(e'))
. . . . |   ⇒
. . . . |   lnk(kind(e)) = lnk(kind(e'))
. . . . |   ⇒
. . . . |   (e <c e'
. . . . |    ⟺
. . . . |    sender(e) <c sender(e')
. . . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . 16. e : E
. . . . 17. l : IdLnk
. . . . 18. n : ℕ||sends(l;e)||
. . . . 19. t' : ℕ
. . . . 20. time(e)+1≤t'
. . . . 21. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . 22. ∃t:ℕ.
. . . . |   isrcv(l;a(destination(l);t))
. . . . |   & ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . ⊢  ∃e':E.
. . . . |  isrcv(kind(e'))
. . . . |  & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. . . . | by Analyze -1
. . . . |
. . . . 1. the_w : World
. . . . 2. FairFifo
. . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . 5. ∀e:E.
. . . . |  ¬first(e)
. . . . |  ⇒
. . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . . 6. ∀e:E. ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . . 7. Trans e,e':E. e <c e'
. . . . 8. SWellFounded(e <c e')
. . . . 9. ∀e:E.
. . . . |  isrcv(kind(e))
. . . . |  ⇒
. . . . |  sends(lnk(kind(e));sender(e))[index(e)]
```

```
. . . . |  =
. . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . 11. ∀e,e':E.
. . . . |   e <c e'
. . . . |   ⇒
. . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . . . 12. ∀e:E. isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . 14. ∀e,e':E.
. . . . |   isrcv(kind(e))
. . . . |   ⇒
. . . . |   isrcv(kind(e'))
. . . . |   ⇒
. . . . |   lnk(kind(e)) = lnk(kind(e'))
. . . . |   ⇒
. . . . |   (e <c e'
. . . . |    ⟺
. . . . |    sender(e) <c sender(e')
. . . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . 16. e : E
. . . . 17. l : IdLnk
. . . . 18. n : ℕ||sends(l;e)||
. . . . 19. t' : ℕ
. . . . 20. time(e)+1≤t'
. . . . 21. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . 22. t : ℕ
. . . . 23. isrcv(l;a(destination(l);t))
. . . . |   & ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . ⊢  ∃e':E.
. . . . |  isrcv(kind(e'))
. . . . |  & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. . . . | by Assert (<destination(l),t> ∈ E)
. . . . |
. . . . |\
. . . . | .....assertion .....
. . . . | 1. the_w : World
. . . . | 2. FairFifo
. . . . | 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . | 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . | 5. ∀e:E.
. . . . |    ¬first(e)
. . . . |    ⇒
. . . . |    loc(pred(e)) = loc(e) & pred(e) <c e
. . . . |    & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . . | 6. ∀e:E.
. . . . |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . . | 7. Trans e,e':E. e <c e'
. . . . | 8. SWellFounded(e <c e')
. . . . | 9. ∀e:E.
. . . . |    isrcv(kind(e))
. . . . |    ⇒
```

```
. . . . |    sends(lnk(kind(e));sender(e))[index(e)]
. . . . |    =
. . . . |    msg(lnk(kind(e));tag(kind(e));val(e))
. . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . | 11. ∀e,e':E.
. . . . |     e <c e'
. . . . |     ⇒
. . . . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . |     ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . . . | 12. ∀e:E.
. . . . |     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . | 14. ∀e,e':E.
. . . . |     isrcv(kind(e))
. . . . |     ⇒
. . . . |     isrcv(kind(e'))
. . . . |     ⇒
. . . . |     lnk(kind(e)) = lnk(kind(e'))
. . . . |     ⇒
. . . . |     (e <c e'
. . . . |      ⟺
. . . . |      sender(e) <c sender(e')
. . . . |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . |     m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . | 16. e : E
. . . . | 17. l : IdLnk
. . . . | 18. n : ℕ||sends(l;e)||
. . . . | 19. t' : ℕ
. . . . | 20. time(e)+1≤t'
. . . . | 21. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . | 22. t : ℕ
. . . . | 23. isrcv(l;a(destination(l);t))
. . . . |     & ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . | ⊢  <destination(l),t> ∈ E
. . . . |   by Analyze THEN Reduce 0 THEN Analyze 0
. . . . |      THEN
. . . . |      MoveToConcl -3
. . . . |      THEN
. . . . |      Unfold `w-isrcvl` 0
. . . . |      THEN
. . . . |      AutoBoolCase isnull(a(destination(l);t))
. . . . |
. . . . .\
. . . . . 1. the_w : World
. . . . . 2. FairFifo
. . . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . 5. ∀e:E.
. . . . . |  ¬first(e)
. . . . . |  ⇒
. . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . . . 6. ∀e:E.
. . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
```

```
. . . . . 7. Trans e,e':E. e <c e'
. . . . . 8. SWellFounded(e <c e')
. . . . . 9. ∀e:E.
. . . . . |  isrcv(kind(e))
. . . . . |  ⇒
. . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . |  =
. . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . 11. ∀e,e':E.
. . . . . |   e <c e'
. . . . . |   ⇒
. . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . . . . 12. ∀e:E.
. . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . 14. ∀e,e':E.
. . . . . |   isrcv(kind(e))
. . . . . |   ⇒
. . . . . |   isrcv(kind(e'))
. . . . . |   ⇒
. . . . . |   lnk(kind(e)) = lnk(kind(e'))
. . . . . |   ⇒
. . . . . |   (e <c e'
. . . . . |    ⟺
. . . . . |    sender(e) <c sender(e')
. . . . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . 16. e : E
. . . . . 17. l : IdLnk
. . . . . 18. n : ℕ||sends(l;e)||
. . . . . 19. t' : ℕ
. . . . . 20. time(e)+1≤t'
. . . . . 21. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . 22. t : ℕ
. . . . . 23. isrcv(l;a(destination(l);t))
. . . . . |   & ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . 24. <destination(l),t> ∈ E
. . . . . ⊢  ∃e':E.
. . . . . |  isrcv(kind(e'))
. . . . . |  & lnk(kind(e')) = l & sender(e') = e & index(e') = n
. . . . . | by InstConcl [<destination(l),t>] THEN MoveToConcl -2
. . . . . |    THEN
. . . . . |    Unfold `w-ekind` 0
. . . . . |    THEN
. . . . . |    Unfold `w-act` 0
. . . . . |    THEN
. . . . . |    Reduce 0
. . . . . |    THEN
. . . . . |    MoveToConcl -3
. . . . . |    THEN
. . . . . |    Unfold `w-isrcvl` 0
. . . . . |    THEN
```

```
. . . . . |    AutoBoolCase isnull(a(destination(l);t))
. . . . . |    THEN
. . . . . |    AutoBoolCase isrcv(kind(a(destination(l);t)))
. . . . . |
. . . . . 1. the_w : World
. . . . . 2. FairFifo
. . . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . 5. ∀e:E.
. . . . . |  ¬first(e)
. . . . . |  ⇒
. . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . |  & (∀e':E. loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . . . 6. ∀e:E.
. . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . . . 7. Trans e,e':E. e <c e'
. . . . . 8. SWellFounded(e <c e')
. . . . . 9. ∀e:E.
. . . . . |  isrcv(kind(e))
. . . . . |  ⇒
. . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . |  =
. . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . 11. ∀e,e':E.
. . . . . |   e <c e'
. . . . . |   ⇒
. . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . . . . 12. ∀e:E.
. . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . 14. ∀e,e':E.
. . . . . |   isrcv(kind(e))
. . . . . |   ⇒
. . . . . |   isrcv(kind(e'))
. . . . . |   ⇒
. . . . . |   lnk(kind(e)) = lnk(kind(e'))
. . . . . |   ⇒
. . . . . |   (e <c e'
. . . . . |    ⟺
. . . . . |    sender(e) <c sender(e')
. . . . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . 16. e : E
. . . . . 17. l : IdLnk
. . . . . 18. n : ℕ||sends(l;e)||
. . . . . 19. t' : ℕ
. . . . . 20. time(e)+1≤t'
. . . . . 21. t : ℕ
. . . . . 22. <destination(l),t> ∈ E
. . . . . 23. ¬isnull(a(destination(l);t))
. . . . . 24. isrcv(kind(a(destination(l);t)))
. . . . . ⊢  ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
```

```
. . . . . |  ⇒
. . . . . |  lnk(kind(a(destination(l);t))) = l
. . . . . |  & ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . |  ⇒
. . . . . |  True
. . . . . |  & lnk(kind(a(destination(l);t))) = l
. . . . . |    & sender(<destination(l),t>) = e
. . . . . |    & index(<destination(l),t>) = n
. . . . . | by Unfold ‘and‘ 0 THEN Fold ‘cand‘ 0
. . . . . |    THEN
. . . . . |    RepeatFor 3 (Analyze 0)
. . . . . |    THEN
. . . . . |    Try Trivial
. . . . . |    THEN
. . . . . |    ExRepD
. . . . . |    THEN
. . . . . |    Analyze 0
. . . . . |
. . . . . |\
. . . . . | 1. the_w : World
. . . . . | 2. FairFifo
. . . . . | 3. ∀e,e’:E. loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
. . . . . | 4. ∀e:E. first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
. . . . . | 5. ∀e:E.
. . . . . |    ¬first(e)
. . . . . |    ⇒
. . . . . |    loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . |    & (∀e’:E.
. . . . . |       loc(e’) = loc(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
. . . . . | 6. ∀e:E.
. . . . . |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . . . | 7. Trans e,e’:E. e <c e’
. . . . . | 8. SWellFounded(e <c e’)
. . . . . | 9. ∀e:E.
. . . . . |    isrcv(kind(e))
. . . . . |    ⇒
. . . . . |    sends(lnk(kind(e));sender(e))[index(e)]
. . . . . |    =
. . . . . |    msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . | 11. ∀e,e’:E.
. . . . . |     e <c e’
. . . . . |     ⇒
. . . . . |     ¬first(e’) & e <c pred(e’) ∨ e = pred(e’)
. . . . . |     ∨ isrcv(kind(e’)) & e <c sender(e’) ∨ e = sender(e’)
. . . . . | 12. ∀e:E.
. . . . . |     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . . . | 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . | 14. ∀e,e’:E.
. . . . . |     isrcv(kind(e))
. . . . . |     ⇒
. . . . . |     isrcv(kind(e’))
. . . . . |     ⇒
. . . . . |     lnk(kind(e)) = lnk(kind(e’))
. . . . . |     ⇒
```

```
. . . . . . |     (e <c e'
. . . . . . |      ⟺
. . . . . . |      sender(e) <c sender(e')
. . . . . . |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . |     m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . | 16. e : E
. . . . . . | 17. l : IdLnk
. . . . . . | 18. n : ℕ||sends(l;e)||
. . . . . . | 19. t' : ℕ
. . . . . . | 20. time(e)+1≤t'
. . . . . . | 21. t : ℕ
. . . . . . | 22. <destination(l),t> ∈ E
. . . . . . | 23. ¬isnull(a(destination(l);t))
. . . . . . | 24. isrcv(kind(a(destination(l);t)))
. . . . . . | 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . | 26. lnk(kind(a(destination(l);t))) = l
. . . . . . | 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . | 28. True
. . . . . . | ⊢  lnk(kind(a(destination(l);t))) = l
. . . . . . |   by RW assert_pushdownC -3
. . . . . . |
. . . . . . .\
. . . . . . . 1. the_w : World
. . . . . . . 2. FairFifo
. . . . . . . 3. ∀e,e':E. loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . 4. ∀e:E. first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . 5. ∀e:E.
. . . . . . . |  ¬first(e)
. . . . . . . |  ⇒
. . . . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . |  & (∀e':E.
. . . . . . . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . . . . . 6. ∀e:E.
. . . . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . 7. Trans e,e':E. e <c e'
. . . . . . . 8. SWellFounded(e <c e')
. . . . . . . 9. ∀e:E.
. . . . . . . |  isrcv(kind(e))
. . . . . . . |  ⇒
. . . . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . |  =
. . . . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . 11. ∀e,e':E.
. . . . . . . |   e <c e'
. . . . . . . |   ⇒
. . . . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . . . . . . 12. ∀e:E.
. . . . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . . . . . 13. ∀e:E, l:IdLnk. ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . 14. ∀e,e':E.
. . . . . . . |   isrcv(kind(e))
. . . . . . . |   ⇒
```

```
. . . . . . . |   isrcv(kind(e'))
. . . . . . . |   ⇒
. . . . . . . |   lnk(kind(e)) = lnk(kind(e'))
. . . . . . . |   ⇒
. . . . . . . |   (e <c e'
. . . . . . . |    ⟺
. . . . . . . |    sender(e) <c sender(e')
. . . . . . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . 16. e : E
. . . . . . . 17. l : IdLnk
. . . . . . . 18. n : ℕ||sends(l;e)||
. . . . . . . 19. t' : ℕ
. . . . . . . 20. time(e)+1≤t'
. . . . . . . 21. t : ℕ
. . . . . . . 22. <destination(l),t> ∈ E
. . . . . . . 23. ¬isnull(a(destination(l);t))
. . . . . . . 24. isrcv(kind(a(destination(l);t)))
. . . . . . . 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . . 26. lnk(kind(a(destination(l);t))) = l
. . . . . . . 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . 28. True
. . . . . . . 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . ⊢  sender(<destination(l),t>) = e
. . . . . . . |  & index(<destination(l),t>) = n
. . . . . . . | by Assert isrcv(kind(<destination(l),t>))
. . . . . . . |
. . . . . . . |\
. . . . . . . | .....assertion .....
. . . . . . . | 1. the_w : World
. . . . . . . | 2. FairFifo
. . . . . . . | 3. ∀e,e':E.
. . . . . . . |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . | 4. ∀e:E.
. . . . . . . |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . | 5. ∀e:E.
. . . . . . . |    ¬first(e)
. . . . . . . |    ⇒
. . . . . . . |    loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . |    & (∀e':E.
. . . . . . . |        loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . . . . . | 6. ∀e:E.
. . . . . . . |    ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . | 7. Trans e,e':E. e <c e'
. . . . . . . | 8. SWellFounded(e <c e')
. . . . . . . | 9. ∀e:E.
. . . . . . . |    isrcv(kind(e))
. . . . . . . |    ⇒
. . . . . . . |    sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . |    =
. . . . . . . |    msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . | 11. ∀e,e':E.
. . . . . . . |     e <c e'
```

```
. . . . . . . |     ⇒
. . . . . . . |     ¬first(e’) & e <c pred(e’) ∨ e = pred(e’)
. . . . . . . |     ∨ isrcv(kind(e’)) & e <c sender(e’) ∨ e = sender(e’)
. . . . . . . | 12. ∀e:E.
. . . . . . . |     isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . . . . . | 13. ∀e:E, l:IdLnk.
. . . . . . . |     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . | 14. ∀e,e’:E.
. . . . . . . |     isrcv(kind(e))
. . . . . . . |     ⇒
. . . . . . . |     isrcv(kind(e’))
. . . . . . . |     ⇒
. . . . . . . |     lnk(kind(e)) = lnk(kind(e’))
. . . . . . . |     ⇒
. . . . . . . |     (e <c e’
. . . . . . . |      ⟺
. . . . . . . |      sender(e) <c sender(e’)
. . . . . . . |      ∨ sender(e) = sender(e’) & index(e)<index(e’))
. . . . . . . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . |     m≤||snds(l;t)|| ⇒ (∃t’:ℕ. t≤t’ & m≤||rcvs(l;t’)||)
. . . . . . . | 16. e : E
. . . . . . . | 17. l : IdLnk
. . . . . . . | 18. n : ℕ||sends(l;e)||
. . . . . . . | 19. t’ : ℕ
. . . . . . . | 20. time(e)+1≤t’
. . . . . . . | 21. t : ℕ
. . . . . . . | 22. <destination(l),t> ∈ E
. . . . . . . | 23. ¬isnull(a(destination(l);t))
. . . . . . . | 24. isrcv(kind(a(destination(l);t)))
. . . . . . . | 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t’)||
. . . . . . . | 26. lnk(kind(a(destination(l);t))) = l
. . . . . . . | 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . | 28. True
. . . . . . . | 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . | ⊢  isrcv(kind(<destination(l),t>))
. . . . . . . |   by Unfold ‘w-ekind‘ 0 THEN Unfold ‘w-act‘ 0
. . . . . . . |      THEN
. . . . . . . |      Reduce 0
. . . . . . . |      THEN
. . . . . . . |      Trivial
. . . . . . . |
. . . . . . . .\
. . . . . . . . 1. the_w : World
. . . . . . . . 2. FairFifo
. . . . . . . . 3. ∀e,e’:E.
. . . . . . . . |  loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
. . . . . . . . 4. ∀e:E.
. . . . . . . . |  first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
. . . . . . . . 5. ∀e:E.
. . . . . . . . |  ¬first(e)
. . . . . . . . |  ⇒
. . . . . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . |  & (∀e’:E.
. . . . . . . . |     loc(e’) = loc(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
. . . . . . . . 6. ∀e:E.
```

```
. . . . . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . 7. Trans e,e':E. e <c e'
. . . . . . . . 8. SWellFounded(e <c e')
. . . . . . . . 9. ∀e:E.
. . . . . . . . |  isrcv(kind(e))
. . . . . . . . |  ⇒
. . . . . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . |  =
. . . . . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . 11. ∀e,e':E.
. . . . . . . . |   e <c e'
. . . . . . . . |   ⇒
. . . . . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . . . . . . . 12. ∀e:E.
. . . . . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . . . . . . 13. ∀e:E, l:IdLnk.
. . . . . . . . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . 14. ∀e,e':E.
. . . . . . . . |   isrcv(kind(e))
. . . . . . . . |   ⇒
. . . . . . . . |   isrcv(kind(e'))
. . . . . . . . |   ⇒
. . . . . . . . |   lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . |   ⇒
. . . . . . . . |   (e <c e'
. . . . . . . . |    ⟺
. . . . . . . . |    sender(e) <c sender(e')
. . . . . . . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . 16. e : E
. . . . . . . . 17. l : IdLnk
. . . . . . . . 18. n : ℕ||sends(l;e)||
. . . . . . . . 19. t' : ℕ
. . . . . . . . 20. time(e)+1≤t'
. . . . . . . . 21. t : ℕ
. . . . . . . . 22. <destination(l),t> ∈ E
. . . . . . . . 23. ¬isnull(a(destination(l);t))
. . . . . . . . 24. isrcv(kind(a(destination(l);t)))
. . . . . . . . 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . . . 26. lnk(kind(a(destination(l);t))) = l
. . . . . . . . 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . . 28. True
. . . . . . . . 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . . 30. isrcv(kind(<destination(l),t>))
. . . . . . . . ⊢  sender(<destination(l),t>) = e
. . . . . . . . |  & index(<destination(l),t>) = n
. . . . . . . . | by RWO "assert-w-eq-E-iff<" 0
. . . . . . . . |
. . . . . . . . 1. the_w : World
. . . . . . . . 2. FairFifo
. . . . . . . . 3. ∀e,e':E.
. . . . . . . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
```

```
. . . . . . . . 4. ∀e:E.
. . . . . . . . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . 5. ∀e:E.
. . . . . . . . |  ¬first(e)
. . . . . . . . |  ⇒
. . . . . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . |  & (∀e':E.
. . . . . . . . |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . . . . . . 6. ∀e:E.
. . . . . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . 7. Trans e,e':E. e <c e'
. . . . . . . . 8. SWellFounded(e <c e')
. . . . . . . . 9. ∀e:E.
. . . . . . . . |  isrcv(kind(e))
. . . . . . . . |  ⇒
. . . . . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . |  =
. . . . . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . 11. ∀e,e':E.
. . . . . . . . |   e <c e'
. . . . . . . . |   ⇒
. . . . . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . |   ∨ isrcv(kind(e')) & e <c sender(e') ∨ e = sender(e')
. . . . . . . . 12. ∀e:E.
. . . . . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . . . . . . 13. ∀e:E, l:IdLnk.
. . . . . . . . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . 14. ∀e,e':E.
. . . . . . . . |   isrcv(kind(e))
. . . . . . . . |   ⇒
. . . . . . . . |   isrcv(kind(e'))
. . . . . . . . |   ⇒
. . . . . . . . |   lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . |   ⇒
. . . . . . . . |   (e <c e'
. . . . . . . . |    ⟺
. . . . . . . . |    sender(e) <c sender(e')
. . . . . . . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . 16. e : E
. . . . . . . . 17. l : IdLnk
. . . . . . . . 18. n : ℕ||sends(l;e)||
. . . . . . . . 19. t' : ℕ
. . . . . . . . 20. time(e)+1≤t'
. . . . . . . . 21. t : ℕ
. . . . . . . . 22. <destination(l),t> ∈ E
. . . . . . . . 23. ¬isnull(a(destination(l);t))
. . . . . . . . 24. isrcv(kind(a(destination(l);t)))
. . . . . . . . 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . . . 26. lnk(kind(a(destination(l);t))) = l
. . . . . . . . 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . . 28. True
. . . . . . . . 29. lnk(kind(a(destination(l);t))) = l
```

```
. . . . . . . . 30. isrcv(kind(<destination(l),t>))
. . . . . . . . ⊢  sender(<destination(l),t>) = e
. . . . . . . . |  & index(<destination(l),t>) = n
. . . . . . . . | by Unfold ‘w-index‘ 0 THEN Unfold ‘w-sender‘ 0
. . . . . . . . |    THEN
. . . . . . . . |    Reduce 0
. . . . . . . . |    THEN
. . . . . . . . |    Unfold ‘w-ekind‘ 0
. . . . . . . . |    THEN
. . . . . . . . |    Unfold ‘w-act‘ 0
. . . . . . . . |    THEN
. . . . . . . . |    Reduce 0
. . . . . . . . |    THEN
. . . . . . . . |    Unfold ‘w-eq-E‘ 0
. . . . . . . . |    THEN
. . . . . . . . |    Reduce 0
. . . . . . . . |
. . . . . . . . 1. the_w : World
. . . . . . . . 2. FairFifo
. . . . . . . . 3. ∀e,e’:E.
. . . . . . . . |  loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
. . . . . . . . 4. ∀e:E.
. . . . . . . . |  first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
. . . . . . . . 5. ∀e:E.
. . . . . . . . |  ¬first(e)
. . . . . . . . |  ⇒
. . . . . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . |  & (∀e’:E.
. . . . . . . . |     loc(e’) = loc(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
. . . . . . . . 6. ∀e:E.
. . . . . . . . |  ¬first(e) ⇒ (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . 7. Trans e,e’:E. e <c e’
. . . . . . . . 8. SWellFounded(e <c e’)
. . . . . . . . 9. ∀e:E.
. . . . . . . . |  isrcv(kind(e))
. . . . . . . . |  ⇒
. . . . . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . |  =
. . . . . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . 11. ∀e,e’:E.
. . . . . . . . |   e <c e’
. . . . . . . . |   ⇒
. . . . . . . . |   ¬first(e’) & e <c pred(e’) ∨ e = pred(e’)
. . . . . . . . |   ∨ isrcv(kind(e’)) & e <c sender(e’) ∨ e = sender(e’)
. . . . . . . . 12. ∀e:E.
. . . . . . . . |   isrcv(kind(e)) ⇒ loc(e) = destination(lnk(kind(e)))
. . . . . . . . 13. ∀e:E, l:IdLnk.
. . . . . . . . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . 14. ∀e,e’:E.
. . . . . . . . |   isrcv(kind(e))
. . . . . . . . |   ⇒

. . . . . . . . |   isrcv(kind(e’))
. . . . . . . . |   ⇒
. . . . . . . . |   lnk(kind(e)) = lnk(kind(e’))
```

```
. . . . . . . . |    ⇒
. . . . . . . . |    (e <c e'
. . . . . . . . |     ⟺
. . . . . . . . |     sender(e) <c sender(e')
. . . . . . . . |     ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . |   m≤||snds(l;t)|| ⇒ (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . 16. e : E
. . . . . . . . 17. l : IdLnk
. . . . . . . . 18. n : ℕ||sends(l;e)||
. . . . . . . . 19. t' : ℕ
. . . . . . . . 20. time(e)+1≤t'
. . . . . . . . 21. t : ℕ
. . . . . . . . 22. <destination(l),t> ∈ E
. . . . . . . . 23. ¬isnull(a(destination(l);t))
. . . . . . . . 24. isrcv(kind(a(destination(l);t)))
. . . . . . . . 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . . . 26. lnk(kind(a(destination(l);t))) = l
. . . . . . . . 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . . 28. True
. . . . . . . . 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . . 30. isrcv(kind(<destination(l),t>))
. . . . . . . . ⊢  (source(lnk(kind(a(destination(l);t)))) = 1of(e)∧_2
. . . . . . . . |    (mu(λt@0.
. . . . . . . . |     match(lnk(kind(a(destination(l);t)));t@0;t))=_2 2of(
. . . . . . . . |                                                       e)))
. . . . . . . . |  & ||rcvs(lnk(kind(a(destination(l);t)));t)||
. . . . . . . . |    -||snds(lnk(kind(a(destination(l);t)));mu(λt@0.
. . . . . . . . |     match(lnk(kind(a(destination(l);t)));t@0;t)))||
. . . . . . . . |     =
. . . . . . . . |     n  by Assert match(l;time(e);t)
. . . . . . . . |
. . . . . . . . |\
. . . . . . . . | .....assertion .....
. . . . . . . . | 1. the_w : World
. . . . . . . . | 2. FairFifo
. . . . . . . . | 3. ∀e,e':E.
. . . . . . . . | |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . . | 4. ∀e:E.
. . . . . . . . | |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . | 5. ∀e:E.
. . . . . . . . | |  ¬first(e)
. . . . . . . . | |  ⇒
. . . . . . . . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . | |  & (∀e':E.
. . . . . . . . | |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . . . . . . | 6. ∀e:E.
. . . . . . . . | |  ¬first(e)
. . . . . . . . | |  ⇒
. . . . . . . . | |  (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . | 7. Trans e,e':E. e <c e'
. . . . . . . . | 8. SWellFounded(e <c e')
. . . . . . . . | 9. ∀e:E.
. . . . . . . . | |  isrcv(kind(e))
. . . . . . . . | |  ⇒
```

```
. . . . . . . . | |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . | |  =
. . . . . . . . | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . | 11. ∀e,e':E.
. . . . . . . . | |   e <c e'
. . . . . . . . | |   ⇒
. . . . . . . . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . | |   ∨ isrcv(kind(e'))
. . . . . . . . | |     & e <c sender(e') ∨ e = sender(e')
. . . . . . . . | 12. ∀e:E.
. . . . . . . . | |   isrcv(kind(e))
. . . . . . . . | |   ⇒
. . . . . . . . | |   loc(e) = destination(lnk(kind(e)))
. . . . . . . . | 13. ∀e:E, l:IdLnk.
. . . . . . . . | |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . | 14. ∀e,e':E.
. . . . . . . . | |   isrcv(kind(e))
. . . . . . . . | |   ⇒
. . . . . . . . | |   isrcv(kind(e'))
. . . . . . . . | |   ⇒
. . . . . . . . | |   lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . | |   ⇒
. . . . . . . . | |   (e <c e'
. . . . . . . . | |    ⟺
. . . . . . . . | |    sender(e) <c sender(e')
. . . . . . . . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . | |   m≤||snds(l;t)||
. . . . . . . . | |   ⇒
. . . . . . . . | |   (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . | 16. e : E
. . . . . . . . | 17. l : IdLnk
. . . . . . . . | 18. n : ℕ||sends(l;e)||
. . . . . . . . | 19. t' : ℕ
. . . . . . . . | 20. time(e)+1≤t'
. . . . . . . . | 21. t : ℕ
. . . . . . . . | 22. <destination(l),t> ∈ E
. . . . . . . . | 23. ¬isnull(a(destination(l);t))
. . . . . . . . | 24. isrcv(kind(a(destination(l);t)))
. . . . . . . . | 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . . . | 26. lnk(kind(a(destination(l);t))) = l

. . . . . . . . | 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . . | 28. True
. . . . . . . . | 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . . | 30. isrcv(kind(<destination(l),t>))
. . . . . . . . | ⊢  match(l;time(e);t)
. . . . . . . . | | by RWO
. . . . . . . . | |    Thm* ∀the_w:World, l:IdLnk, t,t':ℕ.
. . . . . . . . | |         match(l;t;t')
. . . . . . . . | |         ⟺
. . . . . . . . | |         ||snds(l;t)||≤||rcvs(l;t')||
. . . . . . . . | |         & ||rcvs(l;t')||<||snds(l;t)||
. . . . . . . . | |           +||onlnk(l;m(source(l);t))||
. . . . . . . . | |    {assert_DASH_w_DASH_match}
```

```
. . . . . . . . | |    0
. . . . . . . . | |
. . . . . . . . | 1. the_w : World
. . . . . . . . | 2. FairFifo
. . . . . . . . | 3. ∀e,e':E.
. . . . . . . . | |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . . | 4. ∀e:E.
. . . . . . . . | |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . | 5. ∀e:E.
. . . . . . . . | |  ¬first(e)
. . . . . . . . | |  ⇒
. . . . . . . . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . | |  & (∀e':E.
. . . . . . . . | |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . . . . . . | 6. ∀e:E.
. . . . . . . . | |  ¬first(e)
. . . . . . . . | |  ⇒
. . . . . . . . | |  (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . | 7. Trans e,e':E. e <c e'
. . . . . . . . | 8. SWellFounded(e <c e')
. . . . . . . . | 9. ∀e:E.
. . . . . . . . | |  isrcv(kind(e))
. . . . . . . . | |  ⇒
. . . . . . . . | |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . | |  =
. . . . . . . . | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . | 11. ∀e,e':E.
. . . . . . . . | |   e <c e'
. . . . . . . . | |   ⇒
. . . . . . . . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . | |   ∨ isrcv(kind(e'))
. . . . . . . . | |     & e <c sender(e') ∨ e = sender(e')
. . . . . . . . | 12. ∀e:E.
. . . . . . . . | |   isrcv(kind(e))
. . . . . . . . | |   ⇒
. . . . . . . . | |   loc(e) = destination(lnk(kind(e)))
. . . . . . . . | 13. ∀e:E, l:IdLnk.
. . . . . . . . | |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . | 14. ∀e,e':E.
. . . . . . . . | |   isrcv(kind(e))
. . . . . . . . | |   ⇒
. . . . . . . . | |   isrcv(kind(e'))
. . . . . . . . | |   ⇒
. . . . . . . . | |   lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . | |   ⇒
. . . . . . . . | |   (e <c e'
. . . . . . . . | |    ⟺
. . . . . . . . | |    sender(e) <c sender(e')
. . . . . . . . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . | |   m≤||snds(l;t)||
. . . . . . . . | |   ⇒
. . . . . . . . | |   (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . | 16. e : E
```

```
. . . . . . . . | 17. l : IdLnk
. . . . . . . . | 18. n : ℕ||sends(l;e)||
. . . . . . . . | 19. t' : ℕ
. . . . . . . . | 20. time(e)+1≤t'
. . . . . . . . | 21. t : ℕ
. . . . . . . . | 22. <destination(l),t> ∈ E
. . . . . . . . | 23. ¬isnull(a(destination(l);t))
. . . . . . . . | 24. isrcv(kind(a(destination(l);t)))
. . . . . . . . | 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . . . | 26. lnk(kind(a(destination(l);t))) = l
. . . . . . . . | 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . . | 28. True
. . . . . . . . | 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . . | 30. isrcv(kind(<destination(l),t>))
. . . . . . . . | ⊢  ||snds(l;time(e))||≤||rcvs(l;t)||
. . . . . . . . | |  & ||rcvs(l;t)||<||snds(l;time(e))||
. . . . . . . . | |    +||onlnk(l;m(source(l);time(e)))||
. . . . . . . . | | by Unfold `w-sends` 18
. . . . . . . . | |    THEN
. . . . . . . . | |    Subst (loc(e) = source(l)) 18
. . . . . . . . | |    THEN
. . . . . . . . | |    Try
. . . . . . . . | |    (BackThru
. . . . . . . . | |     Thm* ∀w:World, t:ℕ, l:IdLnk, i:Id.
. . . . . . . . | |          ||onlnk(l;m(i;t))|| ∈ ℤ
. . . . . . . . | |     {w_DASH_onlnk_DASH_m})
. . . . . . . . | |
. . . . . . . . | .....equality .....
. . . . . . . . | 1. the_w : World
. . . . . . . . | 2. FairFifo
. . . . . . . . | 3. ∀e,e':E.
. . . . . . . . | |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . . | 4. ∀e:E.
. . . . . . . . | |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . | 5. ∀e:E.
. . . . . . . . | |  ¬first(e)
. . . . . . . . | |  ⇒
. . . . . . . . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . | |  & (∀e':E.
. . . . . . . . | |     loc(e') = loc(e) ⇒ ¬(pred(e) <c e' & e' <c e))
. . . . . . . . | 6. ∀e:E.
. . . . . . . . | |  ¬first(e)
. . . . . . . . | |  ⇒
. . . . . . . . | |  (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . | 7. Trans e,e':E. e <c e'
. . . . . . . . | 8. SWellFounded(e <c e')
. . . . . . . . | 9. ∀e:E.
. . . . . . . . | |  isrcv(kind(e))
. . . . . . . . | |  ⇒
. . . . . . . . | |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . | |  =
. . . . . . . . | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . | 11. ∀e,e':E.
. . . . . . . . | |   e <c e'
```

```
. . . . . . . . | |   ⇒
. . . . . . . . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . | |   ∨ isrcv(kind(e'))
. . . . . . . . | |     & e <c sender(e') ∨ e = sender(e')
. . . . . . . . | 12. ∀e:E.
. . . . . . . . | |   isrcv(kind(e))
. . . . . . . . | |   ⇒
. . . . . . . . | |   loc(e) = destination(lnk(kind(e)))
. . . . . . . . | 13. ∀e:E, l:IdLnk.
. . . . . . . . | |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . | 14. ∀e,e':E.
. . . . . . . . | |   isrcv(kind(e))
. . . . . . . . | |   ⇒
. . . . . . . . | |   isrcv(kind(e'))
. . . . . . . . | |   ⇒
. . . . . . . . | |   lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . | |   ⇒
. . . . . . . . | |   (e <c e'
. . . . . . . . | |    ⇐⇒
. . . . . . . . | |    sender(e) <c sender(e')
. . . . . . . . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . | |   m≤||snds(l;t)||
. . . . . . . . | |   ⇒
. . . . . . . . | |   (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . | 16. e : E
. . . . . . . . | 17. l : IdLnk
. . . . . . . . | 18. n : ℕ||onlnk(l;m(loc(e);time(e)))||
. . . . . . . . | 19. t' : ℕ
. . . . . . . . | 20. time(e)+1≤t'
. . . . . . . . | 21. t : ℕ
. . . . . . . . | 22. <destination(l),t> ∈ E
. . . . . . . . | 23. ¬isnull(a(destination(l);t))
. . . . . . . . | 24. isrcv(kind(a(destination(l);t)))
. . . . . . . . | 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . . . | 26. lnk(kind(a(destination(l);t))) = l
. . . . . . . . | 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . . | 28. True
. . . . . . . . | 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . . | 30. isrcv(kind(<destination(l),t>))
. . . . . . . . | ⊢  loc(e) = source(l)
. . . . . . . . | | by DVar 'e' THEN DVar 'e' THEN All Reduce
. . . . . . . . | |    THEN
. . . . . . . . | |    Assert source(l) = e1
. . . . . . . . | |
. . . . . . . . | |\
. . . . . . . . | | .....assertion .....
. . . . . . . . | | 1. the_w : World
. . . . . . . . | | 2. FairFifo
. . . . . . . . | | 3. ∀e,e':E.
. . . . . . . . | | |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . . | | 4. ∀e:E.
. . . . . . . . | | |  first(e) ⇐⇒ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . | | 5. ∀e:E.
. . . . . . . . | | |  ¬first(e)
```

```
. . . . . . . . | | |  ⇒
. . . . . . . . | | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . | | |  & (∀e':E.
. . . . . . . . | | |     loc(e') = loc(e)
. . . . . . . . | | |     ⇒
. . . . . . . . | | |     ¬(pred(e) <c e' & e' <c e))
. . . . . . . . | | 6. ∀e:E.
. . . . . . . . | | |  ¬first(e)
. . . . . . . . | | |  ⇒
. . . . . . . . | | |  (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . | | 7. Trans e,e':E. e <c e'
. . . . . . . . | | 8. SWellFounded(e <c e')
. . . . . . . . | | 9. ∀e:E.
. . . . . . . . | | |  isrcv(kind(e))
. . . . . . . . | | |  ⇒
. . . . . . . . | | |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . | | |  =
. . . . . . . . | | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . | | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . | | 11. ∀e,e':E.
. . . . . . . . | | |   e <c e'
. . . . . . . . | | |   ⇒
. . . . . . . . | | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . | | |   ∨ isrcv(kind(e'))
. . . . . . . . | | |     & e <c sender(e') ∨ e = sender(e')
. . . . . . . . | | 12. ∀e:E.
. . . . . . . . | | |   isrcv(kind(e))
. . . . . . . . | | |   ⇒
. . . . . . . . | | |   loc(e) = destination(lnk(kind(e)))
. . . . . . . . | | 13. ∀e:E, l:IdLnk.
. . . . . . . . | | |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . | | 14. ∀e,e':E.
. . . . . . . . | | |   isrcv(kind(e))
. . . . . . . . | | |   ⇒
. . . . . . . . | | |   isrcv(kind(e'))
. . . . . . . . | | |   ⇒
. . . . . . . . | | |   lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . | | |   ⇒
. . . . . . . . | | |   (e <c e'
. . . . . . . . | | |    ⟺
. . . . . . . . | | |    sender(e) <c sender(e')
. . . . . . . . | | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . | | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . | | |   m≤||snds(l;t)||
. . . . . . . . | | |   ⇒
. . . . . . . . | | |   (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . | | 16. e1 : Id
. . . . . . . . | | 17. e2 : ℕ
. . . . . . . . | | 18. ¬isnull(a(e1;e2))
. . . . . . . . | | 19. l : IdLnk
. . . . . . . . | | 20. n : ℕ||onlnk(l;m(e1;e2))||
. . . . . . . . | | 21. t' : ℕ
. . . . . . . . | | 22. e2+1≤t'
. . . . . . . . | | 23. t : ℕ
. . . . . . . . | | 24. <destination(l),t> ∈ E
```

```
. . . . . . . . | | 25. ¬isnull(a(destination(l);t))
. . . . . . . . | | 26. isrcv(kind(a(destination(l);t)))
. . . . . . . . | | 27. ||snds(l;e2)||+n+1≤||rcvs(l;t')||
. . . . . . . . | | 28. lnk(kind(a(destination(l);t))) = l
. . . . . . . . | | 29. ||rcvs(l;t)|| = ||snds(l;e2)||+n
. . . . . . . . | | 30. True
. . . . . . . . | | 31. lnk(kind(a(destination(l);t))) = l
. . . . . . . . | | 32. isrcv(kind(<destination(l),t>))
. . . . . . . . | | ⊢  source(l) = e1
. . . . . . . . | | | by SupposeNot THEN RW assert_pushdownC -1
. . . . . . . . | | |    THEN
. . . . . . . . | | |    Analyze 2
. . . . . . . . | | |    THEN
. . . . . . . . | | |    InstHyp [e1;e2;l] 2
. . . . . . . . | | |
. . . . . . . . | | 1. the_w : World
. . . . . . . . | | 2. ∀i:Id, t:ℕ, l:IdLnk.
. . . . . . . . | |    ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. . . . . . . . | | 3. (∀i:Id, t:ℕ.
. . . . . . . . | |     isnull(a(i;t))
. . . . . . . . | |     ⇒
. . . . . . . . | |     (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil)
. . . . . . . . | |    & (∀i:Id, t:ℕ, l:IdLnk.
. . . . . . . . | |       isrcv(l;a(i;t))
. . . . . . . . | |       ⇒
. . . . . . . . | |       destination(l) = i
. . . . . . . . | |       & ||queue(l;t)||≥1
. . . . . . . . | |         & hd(queue(l;t)) = msg(a(i;t)))
. . . . . . . . | |    & (∀l:IdLnk, t:ℕ.
. . . . . . . . | |       ∃t':ℕ.
. . . . . . . . | |       t≤t'
. . . . . . . . | |       & isrcv(l;a(destination(l);t'))
. . . . . . . . | |         ∨ queue(l;t') = nil)
. . . . . . . . | | 4. ∀e,e':E.
. . . . . . . . | |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . . | | 5. ∀e:E.
. . . . . . . . | |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . | | 6. ∀e:E.
. . . . . . . . | |    ¬first(e)
. . . . . . . . | |    ⇒
. . . . . . . . | |    loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . | |    & (∀e':E.
. . . . . . . . | |       loc(e') = loc(e)
. . . . . . . . | |       ⇒
. . . . . . . . | |       ¬(pred(e) <c e' & e' <c e))
. . . . . . . . | | 7. ∀e:E.
. . . . . . . . | |    ¬first(e)
. . . . . . . . | |    ⇒
. . . . . . . . | |    (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . | | 8. Trans e,e':E. e <c e'
. . . . . . . . | | 9. SWellFounded(e <c e')
. . . . . . . . | | 10. ∀e:E.
. . . . . . . . | |     isrcv(kind(e))
. . . . . . . . | |     ⇒
. . . . . . . . | |     sends(lnk(kind(e));sender(e))[index(e)]
```

```
. . . . . . . . | |     =
. . . . . . . . | |     msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . | | 11. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . | | 12. ∀e,e':E.
. . . . . . . . | |     e <c e'
. . . . . . . . | |     ⇒
. . . . . . . . | |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . | |     ∨ isrcv(kind(e'))
. . . . . . . . | |       & e <c sender(e') ∨ e = sender(e')
. . . . . . . . | | 13. ∀e:E.
. . . . . . . . | |     isrcv(kind(e))
. . . . . . . . | |     ⇒
. . . . . . . . | |     loc(e) = destination(lnk(kind(e)))
. . . . . . . . | | 14. ∀e:E, l:IdLnk.
. . . . . . . . | |     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . | | 15. ∀e,e':E.
. . . . . . . . | |     isrcv(kind(e))
. . . . . . . . | |     ⇒
. . . . . . . . | |     isrcv(kind(e'))
. . . . . . . . | |     ⇒
. . . . . . . . | |     lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . | |     ⇒
. . . . . . . . | |     (e <c e'
. . . . . . . . | |      ⟺
. . . . . . . . | |      sender(e) <c sender(e')
. . . . . . . . | |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . | | 16. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . | |     m≤||snds(l;t)||
. . . . . . . . | |     ⇒
. . . . . . . . | |     (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . | | 17. e1 : Id
. . . . . . . . | | 18. e2 : ℕ
. . . . . . . . | | 19. ¬isnull(a(e1;e2))
. . . . . . . . | | 20. l : IdLnk
. . . . . . . . | | 21. n : ℕ||onlnk(l;m(e1;e2))||
. . . . . . . . | | 22. t' : ℕ
. . . . . . . . | | 23. e2+1≤t'
. . . . . . . . | | 24. t : ℕ
. . . . . . . . | | 25. <destination(l),t> ∈ E
. . . . . . . . | | 26. ¬isnull(a(destination(l);t))
. . . . . . . . | | 27. isrcv(kind(a(destination(l);t)))
. . . . . . . . | | 28. ||snds(l;e2)||+n+1≤||rcvs(l;t')||
. . . . . . . . | | 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . . | | 30. ||rcvs(l;t)|| = ||snds(l;e2)||+n
. . . . . . . . | | 31. True
. . . . . . . . | | 32. lnk(kind(a(destination(l);t))) = l
. . . . . . . . | | 33. isrcv(kind(<destination(l),t>))
. . . . . . . . | | 34. ¬source(l) = e1
. . . . . . . . | | 35. onlnk(l;m(e1;e2)) = nil
. . . . . . . . | | ⊢  source(l) = e1  by HypSubst -1 21 THEN Reduce 21
. . . . . . . . | |
. . . . . . . . | .\
. . . . . . . . | . 1. the_w : World
. . . . . . . . | . 2. FairFifo
. . . . . . . . | . 3. ∀e,e':E.
```

```
. . . . . . . . | .    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . . | . 4. ∀e:E.
. . . . . . . . | .    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . | . 5. ∀e:E.
. . . . . . . . | .    ¬first(e)
. . . . . . . . | .    ⇒
. . . . . . . . | .    loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . | .    & (∀e':E.
. . . . . . . . | .       loc(e') = loc(e)
. . . . . . . . | .       ⇒
. . . . . . . . | .       ¬(pred(e) <c e' & e' <c e))
. . . . . . . . | . 6. ∀e:E.
. . . . . . . . | .    ¬first(e)
. . . . . . . . | .    ⇒
. . . . . . . . | .    (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . | . 7. Trans e,e':E. e <c e'
. . . . . . . . | . 8. SWellFounded(e <c e')
. . . . . . . . | . 9. ∀e:E.
. . . . . . . . | .    isrcv(kind(e))
. . . . . . . . | .    ⇒
. . . . . . . . | .    sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . | .    =
. . . . . . . . | .    msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . | . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . | . 11. ∀e,e':E.
. . . . . . . . | .     e <c e'
. . . . . . . . | .     ⇒
. . . . . . . . | .     ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . | .     ∨ isrcv(kind(e'))
. . . . . . . . | .       & e <c sender(e') ∨ e = sender(e')
. . . . . . . . | . 12. ∀e:E.
. . . . . . . . | .     isrcv(kind(e))
. . . . . . . . | .     ⇒
. . . . . . . . | .     loc(e) = destination(lnk(kind(e)))
. . . . . . . . | . 13. ∀e:E, l:IdLnk.
. . . . . . . . | .     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . | . 14. ∀e,e':E.
. . . . . . . . | .     isrcv(kind(e))
. . . . . . . . | .     ⇒
. . . . . . . . | .     isrcv(kind(e'))
. . . . . . . . | .     ⇒
. . . . . . . . | .     lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . | .     ⇒
. . . . . . . . | .     (e <c e'
. . . . . . . . | .      ⟺
. . . . . . . . | .      sender(e) <c sender(e')
. . . . . . . . | .      ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . | . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . | .     m≤||snds(l;t)||
. . . . . . . . | .     ⇒
. . . . . . . . | .     (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . | . 16. e1 : Id
. . . . . . . . | . 17. e2 : ℕ
. . . . . . . . | . 18. ¬isnull(a(e1;e2))
. . . . . . . . | . 19. l : IdLnk
```

```
. . . . . . . . | . 20. n : ℕ||onlnk(l;m(e1;e2))||
. . . . . . . . | . 21. t’ : ℕ
. . . . . . . . | . 22. e2+1≤t’
. . . . . . . . | . 23. t : ℕ
. . . . . . . . | . 24. <destination(l),t> ∈ E
. . . . . . . . | . 25. ¬isnull(a(destination(l);t))
. . . . . . . . | . 26. isrcv(kind(a(destination(l);t)))
. . . . . . . . | . 27. ||snds(l;e2)||+n+1≤||rcvs(l;t’)||
. . . . . . . . | . 28. lnk(kind(a(destination(l);t))) = l
. . . . . . . . | . 29. ||rcvs(l;t)|| = ||snds(l;e2)||+n
. . . . . . . . | . 30. True
. . . . . . . . | . 31. lnk(kind(a(destination(l);t))) = l
. . . . . . . . | . 32. isrcv(kind(<destination(l),t>))
. . . . . . . . | . 33. source(l) = e1
. . . . . . . . | . ⊢  e1 = source(l)  by RW assert_pushdownC -1
. . . . . . . . |
. . . . . . . . .\
. . . . . . . . . 1. the_w : World
. . . . . . . . . 2. FairFifo
. . . . . . . . . 3. ∀e,e’:E.
. . . . . . . . . |  loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
. . . . . . . . . 4. ∀e:E.
. . . . . . . . . |  first(e) ⇔ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
. . . . . . . . . 5. ∀e:E.
. . . . . . . . . |  ¬first(e)
. . . . . . . . . |  ⇒
. . . . . . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . . |  & (∀e’:E.
. . . . . . . . . |     loc(e’) = loc(e) ⇒ ¬(pred(e) <c e’ & e’ <c e))
. . . . . . . . . 6. ∀e:E.
. . . . . . . . . |  ¬first(e)
. . . . . . . . . |  ⇒
. . . . . . . . . |  (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . . 7. Trans e,e’:E. e <c e’
. . . . . . . . . 8. SWellFounded(e <c e’)
. . . . . . . . . 9. ∀e:E.
. . . . . . . . . |  isrcv(kind(e))
. . . . . . . . . |  ⇒
. . . . . . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . . |  =
. . . . . . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . . 11. ∀e,e’:E.
. . . . . . . . . |   e <c e’
. . . . . . . . . |   ⇒
. . . . . . . . . |   ¬first(e’) & e <c pred(e’) ∨ e = pred(e’)
. . . . . . . . . |   ∨ isrcv(kind(e’))
. . . . . . . . . |     & e <c sender(e’) ∨ e = sender(e’)
. . . . . . . . . 12. ∀e:E.
. . . . . . . . . |   isrcv(kind(e))
. . . . . . . . . |   ⇒
. . . . . . . . . |   loc(e) = destination(lnk(kind(e)))
. . . . . . . . . 13. ∀e:E, l:IdLnk.
. . . . . . . . . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . . 14. ∀e,e’:E.
```

```
. . . . . . . . . |   isrcv(kind(e))
. . . . . . . . . |   ⇒
. . . . . . . . . |   isrcv(kind(e'))
. . . . . . . . . |   ⇒
. . . . . . . . . |   lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . . |   ⇒
. . . . . . . . . |   (e <c e'
. . . . . . . . . |    ⟺
. . . . . . . . . |    sender(e) <c sender(e')
. . . . . . . . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . . |   m≤||snds(l;t)||
. . . . . . . . . |   ⇒
. . . . . . . . . |   (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . . 16. e : E
. . . . . . . . . 17. l : IdLnk
. . . . . . . . . 18. n : ℕ||sends(l;e)||
. . . . . . . . . 19. t' : ℕ
. . . . . . . . . 20. time(e)+1≤t'
. . . . . . . . . 21. t : ℕ
. . . . . . . . . 22. <destination(l),t> ∈ E
. . . . . . . . . 23. ¬isnull(a(destination(l);t))
. . . . . . . . . 24. isrcv(kind(a(destination(l);t)))
. . . . . . . . . 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . . . . 26. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . . . 28. True
. . . . . . . . . 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . 30. isrcv(kind(<destination(l),t>))
. . . . . . . . . 31. match(l;time(e);t)
. . . . . . . . . ⊢  (source(lnk(kind(a(destination(l);t)))) = 1of(e)∧₂
. . . . . . . . . |   ...)
. . . . . . . . . |  & ||rcvs(lnk(kind(a(destination(l);t)));t)||
. . . . . . . . . |    -||snds(lnk(kind(a(destination(l);t)));mu(λt@0.
. . . . . . . . . |    match(lnk(kind(a(destination(l);t)));t@0;t)))||
. . . . . . . . . |    =
. . . . . . . . . |    n
. . . . . . . . . | by Subst'
. . . . . . . . . |    (mu(λt@0.
. . . . . . . . . |     match(lnk(kind(a(destination(l);t)));t@0;t))
. . . . . . . . . |     =
. . . . . . . . . |     time(e))
. . . . . . . . . |    0
. . . . . . . . . |
. . . . . . . . . |\
. . . . . . . . . | 1. the_w : World
. . . . . . . . . | 2. FairFifo
. . . . . . . . . | 3. ∀e,e':E.
. . . . . . . . . |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . . . | 4. ∀e:E.
. . . . . . . . . |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . . | 5. ∀e:E.
. . . . . . . . . |    ¬first(e)
. . . . . . . . . |    ⇒
. . . . . . . . . |    loc(pred(e)) = loc(e) & pred(e) <c e
```

```
. . . . . . . . . |    & (∀e':E.
. . . . . . . . . |       loc(e') = loc(e)
. . . . . . . . . |       ⇒
. . . . . . . . . |       ¬(pred(e) <c e' & e' <c e))
. . . . . . . . . | 6. ∀e:E.
. . . . . . . . . |    ¬first(e)
. . . . . . . . . |    ⇒
. . . . . . . . . |    (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . . | 7. Trans e,e':E. e <c e'
. . . . . . . . . | 8. SWellFounded(e <c e')
. . . . . . . . . | 9. ∀e:E.
. . . . . . . . . |    isrcv(kind(e))
. . . . . . . . . |    ⇒
. . . . . . . . . |    sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . . |    =
. . . . . . . . . |    msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . . | 11. ∀e,e':E.
. . . . . . . . . |     e <c e'
. . . . . . . . . |     ⇒
. . . . . . . . . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . . |     ∨ isrcv(kind(e'))
. . . . . . . . . |       & e <c sender(e') ∨ e = sender(e')
. . . . . . . . . | 12. ∀e:E.
. . . . . . . . . |     isrcv(kind(e))
. . . . . . . . . |     ⇒
. . . . . . . . . |     loc(e) = destination(lnk(kind(e)))
. . . . . . . . . | 13. ∀e:E, l:IdLnk.
. . . . . . . . . |     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . . | 14. ∀e,e':E.
. . . . . . . . . |     isrcv(kind(e))
. . . . . . . . . |     ⇒
. . . . . . . . . |     isrcv(kind(e'))
. . . . . . . . . |     ⇒
. . . . . . . . . |     lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . . |     ⇒
. . . . . . . . . |     (e <c e'
. . . . . . . . . |      ⟺
. . . . . . . . . |      sender(e) <c sender(e')
. . . . . . . . . |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . . |     m≤||snds(l;t)||
. . . . . . . . . |     ⇒
. . . . . . . . . |     (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . . | 16. e : E
. . . . . . . . . | 17. l : IdLnk
. . . . . . . . . | 18. n : ℕ||sends(l;e)||
. . . . . . . . . | 19. t' : ℕ
. . . . . . . . . | 20. time(e)+1≤t'
. . . . . . . . . | 21. t : ℕ
. . . . . . . . . | 22. <destination(l),t> ∈ E
. . . . . . . . . | 23. ¬isnull(a(destination(l);t))
. . . . . . . . . | 24. isrcv(kind(a(destination(l);t)))
. . . . . . . . . | 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . . . . | 26. lnk(kind(a(destination(l);t))) = l
```

```
. . . . . . . . . | 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . . . | 28. True
. . . . . . . . . | 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . | 30. isrcv(kind(<destination(l),t>))
. . . . . . . . . | 31. match(l;time(e);t)
. . . . . . . . . | ⊢  ∃n:ℕ.
. . . . . . . . . |    (λt@0.
. . . . . . . . . |     match(lnk(kind(a(destination(l);t)));t@0;t))
. . . . . . . . . |    (n)
. . . . . . . . . |   by Reduce 0 THEN InstConcl [time(e)]
. . . . . . . . . |      THEN
. . . . . . . . . |      HypSubst -3 0
. . . . . . . . . |
. . . . . . . . . |\
. . . . . . . . . | 1. the_w : World
. . . . . . . . . | 2. FairFifo
. . . . . . . . . | 3. ∀e,e':E.
. . . . . . . . . | |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . . . | 4. ∀e:E.
. . . . . . . . . | |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . . | 5. ∀e:E.
. . . . . . . . . | |  ¬first(e)
. . . . . . . . . | |  ⇒
. . . . . . . . . | |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . . | |  & (∀e':E.
. . . . . . . . . | |     loc(e') = loc(e)
. . . . . . . . . | |     ⇒
. . . . . . . . . | |     ¬(pred(e) <c e' & e' <c e))
. . . . . . . . . | 6. ∀e:E.
. . . . . . . . . | |  ¬first(e)
. . . . . . . . . | |  ⇒
. . . . . . . . . | |  (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . . | 7. Trans e,e':E. e <c e'
. . . . . . . . . | 8. SWellFounded(e <c e')
. . . . . . . . . | 9. ∀e:E.
. . . . . . . . . | |  isrcv(kind(e))
. . . . . . . . . | |  ⇒
. . . . . . . . . | |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . . | |  =
. . . . . . . . . | |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . . | 11. ∀e,e':E.
. . . . . . . . . | |   e <c e'
. . . . . . . . . | |   ⇒
. . . . . . . . . | |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . . | |   ∨ isrcv(kind(e'))
. . . . . . . . . | |     & e <c sender(e') ∨ e = sender(e')
. . . . . . . . . | 12. ∀e:E.
. . . . . . . . . | |   isrcv(kind(e))
. . . . . . . . . | |   ⇒
. . . . . . . . . | |   loc(e) = destination(lnk(kind(e)))
. . . . . . . . . | 13. ∀e:E, l:IdLnk.
. . . . . . . . . | |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . . | 14. ∀e,e':E.
. . . . . . . . . | |   isrcv(kind(e))
```

```
. . . . . . . . . | |   ⇒
. . . . . . . . . | |   isrcv(kind(e'))
. . . . . . . . . | |   ⇒
. . . . . . . . . | |   lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . . | |   ⇒
. . . . . . . . . | |   (e <c e'
. . . . . . . . . | |    ⟺
. . . . . . . . . | |    sender(e) <c sender(e')
. . . . . . . . . | |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . . | |   m≤||snds(l;t)||
. . . . . . . . . | |   ⇒
. . . . . . . . . | |   (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . . | 16. e : E
. . . . . . . . . | 17. l : IdLnk
. . . . . . . . . | 18. n : ℕ||sends(l;e)||
. . . . . . . . . | 19. t' : ℕ
. . . . . . . . . | 20. time(e)+1≤t'
. . . . . . . . . | 21. t : ℕ
. . . . . . . . . | 22. <destination(l),t> ∈ E
. . . . . . . . . | 23. ¬isnull(a(destination(l);t))
. . . . . . . . . | 24. isrcv(kind(a(destination(l);t)))
. . . . . . . . . | 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . . . . | 26. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . | 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . . . | 28. True
. . . . . . . . . | 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . | 30. isrcv(kind(<destination(l),t>))
. . . . . . . . . | 31. match(l;time(e);t)
. . . . . . . . . | ⊢  mu(λt@0.
. . . . . . . . . | |  match(lnk(kind(a(destination(l);t)));t@0;t))
. . . . . . . . . | |  =
. . . . . . . . . | |  time(e)
. . . . . . . . . | | by AssertBY
. . . . . . . . . | |    (∃n:ℕ.
. . . . . . . . . | |     match(lnk(kind(a(destination(l);t)));n;t))
. . . . . . . . . | |    ((HypSubst -3 0) THEN (InstConcl [time(e)]))
. . . . . . . . . | |    THEN
. . . . . . . . . | |    Inst
. . . . . . . . . | |    Thm* ∀f:(ℕ→𝔹).
. . . . . . . . . | |         (∃n:ℕ. f(n))
. . . . . . . . . | |         ⇒
. . . . . . . . . | |         f(mu(f)) & (∀i:ℕ. i<mu(f) ⇒ ¬f(i))
. . . . . . . . . | |    {mu_DASH_property}
. . . . . . . . . | |    [λt@0.
. . . . . . . . . | |     match(lnk(kind(a(destination(l);t)));t@0;t)]
. . . . . . . . . | |    THEN
. . . . . . . . . | |    All Reduce
. . . . . . . . . | |    THEN
. . . . . . . . . | |    Try Trivial
. . . . . . . . . | |
. . . . . . . . . | 1. the_w : World
. . . . . . . . . | 2. FairFifo
. . . . . . . . . | 3. ∀e,e':E.
. . . . . . . . . |    loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
```

```
. . . . . . . . . | 4. ∀e:E.
. . . . . . . . . |    first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . . | 5. ∀e:E.
. . . . . . . . . |    ¬first(e)
. . . . . . . . . |    ⇒
. . . . . . . . . |    loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . . |    & (∀e':E.
. . . . . . . . . |       loc(e') = loc(e)
. . . . . . . . . |       ⇒
. . . . . . . . . |       ¬(pred(e) <c e' & e' <c e))
. . . . . . . . . | 6. ∀e:E.
. . . . . . . . . |    ¬first(e)
. . . . . . . . . |    ⇒
. . . . . . . . . |    (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . . | 7. Trans e,e':E. e <c e'
. . . . . . . . . | 8. SWellFounded(e <c e')
. . . . . . . . . | 9. ∀e:E.
. . . . . . . . . |    isrcv(kind(e))
. . . . . . . . . |    ⇒
. . . . . . . . . |    sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . . |    =
. . . . . . . . . |    msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . . | 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . . | 11. ∀e,e':E.
. . . . . . . . . |     e <c e'
. . . . . . . . . |     ⇒
. . . . . . . . . |     ¬first(e') & e <c pred(e') ∨ e = pred(e')

. . . . . . . . . |     ∨ isrcv(kind(e'))
. . . . . . . . . |       & e <c sender(e') ∨ e = sender(e')
. . . . . . . . . | 12. ∀e:E.
. . . . . . . . . |     isrcv(kind(e))
. . . . . . . . . |     ⇒
. . . . . . . . . |     loc(e) = destination(lnk(kind(e)))
. . . . . . . . . | 13. ∀e:E, l:IdLnk.
. . . . . . . . . |     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . . | 14. ∀e,e':E.
. . . . . . . . . |     isrcv(kind(e))
. . . . . . . . . |     ⇒
. . . . . . . . . |     isrcv(kind(e'))
. . . . . . . . . |     ⇒
. . . . . . . . . |     lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . . |     ⇒
. . . . . . . . . |     (e <c e'
. . . . . . . . . |      ⟺
. . . . . . . . . |      sender(e) <c sender(e')
. . . . . . . . . |      ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . . | 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . . |     m≤||snds(l;t)||
. . . . . . . . . |     ⇒
. . . . . . . . . |     (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . . | 16. e : E
. . . . . . . . . | 17. l : IdLnk
. . . . . . . . . | 18. n : ℕ||sends(l;e)||
. . . . . . . . . | 19. t' : ℕ
. . . . . . . . . | 20. time(e)+1≤t'
```

```
. . . . . . . . . | 21. t : ℕ
. . . . . . . . . | 22. <destination(l),t> ∈ E
. . . . . . . . . | 23. ¬isnull(a(destination(l);t))
. . . . . . . . . | 24. isrcv(kind(a(destination(l);t)))
. . . . . . . . . | 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . . . . | 26. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . | 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . . . | 28. True
. . . . . . . . . | 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . | 30. isrcv(kind(<destination(l),t>))
. . . . . . . . . | 31. match(l;time(e);t)
. . . . . . . . . | 32. ∃n:ℕ. match(lnk(kind(a(destination(l);t)));n;t)
. . . . . . . . . | 33. match(lnk(kind(a(destination(l);t)));mu(λt@0.
. . . . . . . . . |     match(lnk(kind(a(destination(l);t)));t@0;t));t)
. . . . . . . . . |     & (∀i:ℕ.
. . . . . . . . . |        i<mu(λt@0.
. . . . . . . . . |        match(lnk(kind(a(destination(l);t)));t@0;t))
. . . . . . . . . |        ⇒
. . . . . . . . . |        ¬match(lnk(kind(a(destination(l);t)));i;t))
. . . . . . . . . | ⊢  mu(λt@0.
. . . . . . . . . |    match(lnk(kind(a(destination(l);t)));t@0;t))
. . . . . . . . . |    =
. . . . . . . . . |    time(e)
. . . . . . . . . |   by Using [‘the_w’,the_w;‘e’,<destination(l),t>]
. . . . . . . . . |      (BackThru
. . . . . . . . . |       Thm* ∀the_w:World, e:E, t,t':ℕ.
. . . . . . . . . |            FairFifo
. . . . . . . . . |            ⇒
. . . . . . . . . |            isrcv(kind(e))
. . . . . . . . . |            ⇒
. . . . . . . . . |            match(lnk(kind(e));t;time(e))
. . . . . . . . . |            ⇒
. . . . . . . . . |            match(lnk(kind(e));t';time(e)) ⇒ t = t'
. . . . . . . . . |       {w_DASH_match_DASH_unique})
. . . . . . . . . |      THEN
. . . . . . . . . |      Reduce 0
. . . . . . . . . |      THEN
. . . . . . . . . |      Try
. . . . . . . . . |      (Unfold ‘w-ekind‘ 0 THEN Unfold ‘w-act‘ 0
. . . . . . . . . |       THEN
. . . . . . . . . |       Reduce 0)
. . . . . . . . . |      THEN
. . . . . . . . . |      HypSubst -6 0
. . . . . . . . . |
. . . . . . . . .\
. . . . . . . . . . 1. the_w : World
. . . . . . . . . . 2. FairFifo
. . . . . . . . . . 3. ∀e,e':E.
. . . . . . . . . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . . . . 4. ∀e:E.
. . . . . . . . . . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . . . 5. ∀e:E.
. . . . . . . . . . |  ¬first(e)
. . . . . . . . . . |  ⇒
. . . . . . . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
```

```
. . . . . . . . . . |  & (∀e':E.
. . . . . . . . . . |     loc(e') = loc(e)
. . . . . . . . . . |     ⇒
. . . . . . . . . . |     ¬(pred(e) <c e' & e' <c e))
. . . . . . . . . . 6. ∀e:E.
. . . . . . . . . . |  ¬first(e)
. . . . . . . . . . |  ⇒
. . . . . . . . . . |  (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . . . 7. Trans e,e':E. e <c e'
. . . . . . . . . . 8. SWellFounded(e <c e')
. . . . . . . . . . 9. ∀e:E.
. . . . . . . . . . |  isrcv(kind(e))
. . . . . . . . . . |  ⇒
. . . . . . . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . . . |  =
. . . . . . . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e

. . . . . . . . . . 11. ∀e,e':E.
. . . . . . . . . . |   e <c e'
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . . . |   ∨ isrcv(kind(e'))
. . . . . . . . . . |     & e <c sender(e') ∨ e = sender(e')
. . . . . . . . . . 12. ∀e:E.
. . . . . . . . . . |   isrcv(kind(e))
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   loc(e) = destination(lnk(kind(e)))
. . . . . . . . . . 13. ∀e:E, l:IdLnk.
. . . . . . . . . . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . . . 14. ∀e,e':E.
. . . . . . . . . . |   isrcv(kind(e))
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   isrcv(kind(e'))
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   (e <c e'
. . . . . . . . . . |    ⟺
. . . . . . . . . . |    sender(e) <c sender(e')
. . . . . . . . . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . . . |   m≤||snds(l;t)||
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . . . 16. e : E
. . . . . . . . . . 17. l : IdLnk
. . . . . . . . . . 18. n : ℕ||sends(l;e)||
. . . . . . . . . . 19. t' : ℕ
. . . . . . . . . . 20. time(e)+1≤t'
. . . . . . . . . . 21. t : ℕ
. . . . . . . . . . 22. <destination(l),t> ∈ E
. . . . . . . . . . 23. ¬isnull(a(destination(l);t))
. . . . . . . . . . 24. isrcv(kind(a(destination(l);t)))
. . . . . . . . . . 25. ||snds(l;time(e))||+n+1≤||rcvs(l;t')||
. . . . . . . . . . 26. lnk(kind(a(destination(l);t))) = l
```

```
. . . . . . . . . . 27. ||rcvs(l;t)|| = ||snds(l;time(e))||+n
. . . . . . . . . . 28. True
. . . . . . . . . . 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . . 30. isrcv(kind(<destination(l),t>))
. . . . . . . . . . 31. match(l;time(e);t)
. . . . . . . . . . ⊢  (source(lnk(kind(a(destination(l);t)))) = 1of(e)
. . . . . . . . . . |   ∧_2
. . . . . . . . . . |   (time(e)=_2 2of(e)))
. . . . . . . . . . |  & ||rcvs(lnk(kind(a(destination(l);t)));t)||
. . . . . . . . . . |    -||snds(lnk(kind(a(destination(l);t)));...)||
. . . . . . . . . . |    =
. . . . . . . . . . |    n
. . . . . . . . . . | by DVar ‘e’ THEN DVar ‘e’ THEN All Reduce
. . . . . . . . . . |    THEN
. . . . . . . . . . |    HypSubst -3 0
. . . . . . . . . . |
. . . . . . . . . . 1. the_w : World
. . . . . . . . . . 2. FairFifo
. . . . . . . . . . 3. ∀e,e’:E.
. . . . . . . . . . |  loc(e) = loc(e’) ⇒ e <c e’ ∨ e = e’ ∨ e’ <c e
. . . . . . . . . . 4. ∀e:E.
. . . . . . . . . . |  first(e) ⟺ (∀e’:E. loc(e’) = loc(e) ⇒ ¬e’ <c e)
. . . . . . . . . . 5. ∀e:E.
. . . . . . . . . . |  ¬first(e)
. . . . . . . . . . |  ⇒
. . . . . . . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . . . |  & (∀e’:E.
. . . . . . . . . . |     loc(e’) = loc(e)
. . . . . . . . . . |     ⇒
. . . . . . . . . . |     ¬(pred(e) <c e’ & e’ <c e))
. . . . . . . . . . 6. ∀e:E.
. . . . . . . . . . |  ¬first(e)
. . . . . . . . . . |  ⇒
. . . . . . . . . . |  (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . . . 7. Trans e,e’:E. e <c e’
. . . . . . . . . . 8. SWellFounded(e <c e’)
. . . . . . . . . . 9. ∀e:E.
. . . . . . . . . . |  isrcv(kind(e))
. . . . . . . . . . |  ⇒
. . . . . . . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . . . |  =
. . . . . . . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . . . 11. ∀e,e’:E.
. . . . . . . . . . |   e <c e’
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   ¬first(e’) & e <c pred(e’) ∨ e = pred(e’)
. . . . . . . . . . |   ∨ isrcv(kind(e’))
. . . . . . . . . . |     & e <c sender(e’) ∨ e = sender(e’)
. . . . . . . . . . 12. ∀e:E.
. . . . . . . . . . |   isrcv(kind(e))
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   loc(e) = destination(lnk(kind(e)))
. . . . . . . . . . 13. ∀e:E, l:IdLnk.
. . . . . . . . . . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
```

```
. . . . . . . . . . 14. ∀e,e':E.
. . . . . . . . . . |   isrcv(kind(e))
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   isrcv(kind(e'))
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   (e <c e'
. . . . . . . . . . |    ⟺
. . . . . . . . . . |    sender(e) <c sender(e')
. . . . . . . . . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . . . |   m≤||snds(l;t)||
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . . . 16. e1 : Id
. . . . . . . . . . 17. e2 : ℕ
. . . . . . . . . . 18. ¬isnull(a(e1;e2))  [not for witness]
. . . . . . . . . . 19. l : IdLnk
. . . . . . . . . . 20. n : ℕ||sends(l;<e1,e2>)||
. . . . . . . . . . 21. t' : ℕ
. . . . . . . . . . 22. e2+1≤t'
. . . . . . . . . . 23. t : ℕ
. . . . . . . . . . 24. <destination(l),t> ∈ E
. . . . . . . . . . 25. ¬isnull(a(destination(l);t))
. . . . . . . . . . 26. isrcv(kind(a(destination(l);t)))
. . . . . . . . . . 27. ||snds(l;e2)||+n+1≤||rcvs(l;t')||
. . . . . . . . . . 28. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . . 29. ||rcvs(l;t)|| = ||snds(l;e2)||+n
. . . . . . . . . . 30. True
. . . . . . . . . . 31. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . . 32. isrcv(kind(<destination(l),t>))
. . . . . . . . . . 33. match(l;e2;t)
. . . . . . . . . . ⊢  source(l) = e1∧_2(e2=_2e2)
. . . . . . . . . . | by RW assert_pushdownC 0
. . . . . . . . . . |
. . . . . . . . . . 1. the_w : World
. . . . . . . . . . 2. FairFifo
. . . . . . . . . . 3. ∀e,e':E.
. . . . . . . . . . |  loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . . . . 4. ∀e:E.
. . . . . . . . . . |  first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . . . 5. ∀e:E.
. . . . . . . . . . |  ¬first(e)
. . . . . . . . . . |  ⇒
. . . . . . . . . . |  loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . . . |  & (∀e':E.
. . . . . . . . . . |     loc(e') = loc(e)
. . . . . . . . . . |     ⇒
. . . . . . . . . . |     ¬(pred(e) <c e' & e' <c e))
. . . . . . . . . . 6. ∀e:E.
. . . . . . . . . . |  ¬first(e)
. . . . . . . . . . |  ⇒
. . . . . . . . . . |  (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . . . 7. Trans e,e':E. e <c e'
```

```
. . . . . . . . . . 8. SWellFounded(e <c e')
. . . . . . . . . . 9. ∀e:E.
. . . . . . . . . . |  isrcv(kind(e))
. . . . . . . . . . |  ⇒
. . . . . . . . . . |  sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . . . |  =
. . . . . . . . . . |  msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . . . 10. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . . . 11. ∀e,e':E.
. . . . . . . . . . |   e <c e'
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . . . |   ∨ isrcv(kind(e'))
. . . . . . . . . . |     & e <c sender(e') ∨ e = sender(e')
. . . . . . . . . . 12. ∀e:E.
. . . . . . . . . . |   isrcv(kind(e))
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   loc(e) = destination(lnk(kind(e)))
. . . . . . . . . . 13. ∀e:E, l:IdLnk.
. . . . . . . . . . |   ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . . . 14. ∀e,e':E.
. . . . . . . . . . |   isrcv(kind(e))
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   isrcv(kind(e'))
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   (e <c e'
. . . . . . . . . . |    ⟺
. . . . . . . . . . |    sender(e) <c sender(e')
. . . . . . . . . . |    ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . . . 15. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . . . |   m≤||snds(l;t)||
. . . . . . . . . . |   ⇒
. . . . . . . . . . |   (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . . . 16. e1 : Id
. . . . . . . . . . 17. e2 : ℕ
. . . . . . . . . . 18. ¬isnull(a(e1;e2))
. . . . . . . . . . 19. l : IdLnk
. . . . . . . . . . 20. n : ℕ||sends(l;<e1,e2>)||
. . . . . . . . . . 21. t' : ℕ
. . . . . . . . . . 22. e2+1≤t'
. . . . . . . . . . 23. t : ℕ
. . . . . . . . . . 24. <destination(l),t> ∈ E
. . . . . . . . . . 25. ¬isnull(a(destination(l);t))
. . . . . . . . . . 26. isrcv(kind(a(destination(l);t)))
. . . . . . . . . . 27. ||snds(l;e2)||+n+1≤||rcvs(l;t')||
. . . . . . . . . . 28. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . . 29. ||rcvs(l;t)|| = ||snds(l;e2)||+n
. . . . . . . . . . 30. True
. . . . . . . . . . 31. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . . 32. isrcv(kind(<destination(l),t>))

. . . . . . . . . . 33. match(l;e2;t)
. . . . . . . . . . ⊢  source(l) = e1
. . . . . . . . . . | by SupposeNot THEN Analyze 2
```

```
. . . . . . . . . . |    THEN
. . . . . . . . . . |    InstHyp [e1;e2;l] 2
. . . . . . . . . . |
. . . . . . . . . . 1. the_w : World
. . . . . . . . . . 2. ∀i:Id, t:ℕ, l:IdLnk.
. . . . . . . . .      ¬source(l) = i ⇒ onlnk(l;m(i;t)) = nil
. . . . . . . . . . 3. (∀i:Id, t:ℕ.
. . . . . . . . .       isnull(a(i;t))
. . . . . . . . .       ⇒
. . . . . . . . .       (∀x:Id. s(i;t+1).x = s(i;t).x) & m(i;t) = nil)
. . . . . . . . .      & (∀i:Id, t:ℕ, l:IdLnk.
. . . . . . . . .         isrcv(l;a(i;t))
. . . . . . . . .         ⇒
. . . . . . . . .         destination(l) = i
. . . . . . . . .         & ||queue(l;t)||≥1
. . . . . . . . .           & hd(queue(l;t)) = msg(a(i;t)))
. . . . . . . . .      & (∀l:IdLnk, t:ℕ.
. . . . . . . . .         ∃t':ℕ.
. . . . . . . . .         t≤t'
. . . . . . . . .         & isrcv(l;a(destination(l);t'))
. . . . . . . . .           ∨ queue(l;t') = nil)
. . . . . . . . . . 4. ∀e,e':E.
. . . . . . . . .      loc(e) = loc(e') ⇒ e <c e' ∨ e = e' ∨ e' <c e
. . . . . . . . . . 5. ∀e:E.
. . . . . . . . .      first(e) ⟺ (∀e':E. loc(e') = loc(e) ⇒ ¬e' <c e)
. . . . . . . . . . 6. ∀e:E.
. . . . . . . . .      ¬first(e)
. . . . . . . . .      ⇒
. . . . . . . . .      loc(pred(e)) = loc(e) & pred(e) <c e
. . . . . . . . .      & (∀e':E.
. . . . . . . . .         loc(e') = loc(e)
. . . . . . . . .         ⇒
. . . . . . . . .         ¬(pred(e) <c e' & e' <c e))
. . . . . . . . . . 7. ∀e:E.
. . . . . . . . .      ¬first(e)
. . . . . . . . .      ⇒
. . . . . . . . .      (∀x:Id. (x when e) = (x after pred(e)))
. . . . . . . . . . 8. Trans e,e':E. e <c e'
. . . . . . . . . . 9. SWellFounded(e <c e')
. . . . . . . . . . 10. ∀e:E.
. . . . . . . . .       isrcv(kind(e))
. . . . . . . . .       ⇒
. . . . . . . . .       sends(lnk(kind(e));sender(e))[index(e)]
. . . . . . . . .       =
. . . . . . . . .       msg(lnk(kind(e));tag(kind(e));val(e))
. . . . . . . . . . 11. ∀e:E. isrcv(kind(e)) ⇒ sender(e) <c e
. . . . . . . . . . 12. ∀e,e':E.
. . . . . . . . .       e <c e'
. . . . . . . . .       ⇒
. . . . . . . . .       ¬first(e') & e <c pred(e') ∨ e = pred(e')
. . . . . . . . .       ∨ isrcv(kind(e'))
. . . . . . . . .         & e <c sender(e') ∨ e = sender(e')
. . . . . . . . . . 13. ∀e:E.
. . . . . . . . .       isrcv(kind(e))
. . . . . . . . .       ⇒
```

```
. . . . . . . . . .     loc(e) = destination(lnk(kind(e)))
. . . . . . . . . . 14. ∀e:E, l:IdLnk.
. . . . . . . . . .     ¬loc(e) = source(l) ⇒ sends(l;e) = nil
. . . . . . . . . . 15. ∀e,e':E.
. . . . . . . . . .     isrcv(kind(e))
. . . . . . . . . .     ⇒
. . . . . . . . . .     isrcv(kind(e'))
. . . . . . . . . .     ⇒
. . . . . . . . . .     lnk(kind(e)) = lnk(kind(e'))
. . . . . . . . . .     ⇒
. . . . . . . . . .     (e <c e'
. . . . . . . . . .      ⟺
. . . . . . . . . .      sender(e) <c sender(e')
. . . . . . . . . .      ∨ sender(e) = sender(e') & index(e)<index(e'))
. . . . . . . . . . 16. ∀m:ℕ, l:IdLnk, t:ℕ.
. . . . . . . . . .     m≤||snds(l;t)||
. . . . . . . . . .     ⇒
. . . . . . . . . .     (∃t':ℕ. t≤t' & m≤||rcvs(l;t')||)
. . . . . . . . . . 17. e1 : Id
. . . . . . . . . . 18. e2 : ℕ
. . . . . . . . . . 19. ¬isnull(a(e1;e2))
. . . . . . . . . . 20. l : IdLnk
. . . . . . . . . . 21. n : ℕ||sends(l;<e1,e2>)||
. . . . . . . . . . 22. t' : ℕ
. . . . . . . . . . 23. e2+1≤t'
. . . . . . . . . . 24. t : ℕ
. . . . . . . . . . 25. <destination(l),t> ∈ E
. . . . . . . . . . 26. ¬isnull(a(destination(l);t))
. . . . . . . . . . 27. isrcv(kind(a(destination(l);t)))
. . . . . . . . . . 28. ||snds(l;e2)||+n+1≤||rcvs(l;t')||
. . . . . . . . . . 29. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . . 30. ||rcvs(l;t)|| = ||snds(l;e2)||+n
. . . . . . . . . . 31. True
. . . . . . . . . . 32. lnk(kind(a(destination(l);t))) = l
. . . . . . . . . . 33. isrcv(kind(<destination(l),t>))
. . . . . . . . . . 34. match(l;e2;t)
. . . . . . . . . . 35. ¬source(l) = e1
. . . . . . . . . . 36. onlnk(l;m(e1;e2)) = nil
. . . . . . . . . . ⊢  source(l) = e1
. . . . . . . . . .   by Unfold `w-sends` 21 THEN Reduce 21
. . . . . . . . . .      THEN
. . . . . . . . . .      HypSubst -1 21

. . . . . . . . . .      THEN
. . . . . . . . . .      Reduce 25
END
```