```
effect-rule  THM

⊢  ∀i,x:Id, k:Knd, ds:x:Id fp-> Type, da:a:Knd fp-> Type,
|   f:(State(ds)→ma-valtype(da; k)→ds(x)?Void).
|  d-single-effect(i; ds; da; k; x; f)
|  realizes es.(∀x:Id. vartype(i;x) ⊆r ds(x)?Top)
|              & (∀e:E.
|                 loc(e) = i
|                 ⇒
|                 (valtype(e) ⊆r ma-valtype(da; kind(e))))
|              & (∀e:E.
|                 loc(e) = i
|                 ⇒
|                 kind(e) = k
|                 ⇒
|                 (x after e) = f((λz.(z when e)),val(e)))
| by Auto
|    THEN
|    BackThru
|    Thm* ∀D:Dsys, P:({es:ES| es is an event system of D }→Prop{i'}).
|         D realizes2 es.P(es) ⇒ D realizes es.P(es)
|    {d_DASH_realizes2_DASH_implies_DASH_realizes}
|    THENA
|    Try (Complete Auto)
|
|\
| .....wf .....
| 1. i : Id
| 2. x : Id
| 3. k : Knd
| 4. ds : x:Id fp-> Type
| 5. da : a:Knd fp-> Type
| 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
| ⊢  (λes.(∀x:Id. vartype(i;x) ⊆r ds(x)?Top)
| |   & (∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e))))
| |   & (∀e:E.
| |      loc(e) = i
| |      ⇒
| |      kind(e) = k ⇒ (x after e) = f((λz.(z when e)),val(e))))
| |  ∈ {es:ES
| |     | es is an event system of d-single-effect
| |       (i; ds; da; k; x; f) }
| |    →Prop  by Analyze THEN Analyze -1 THEN Analyze -1 THEN ExRepD
| |
| |\
| | 1. i : Id
| | 2. x : Id
| | 3. k : Knd
| | 4. ds : x:Id fp-> Type
| | 5. da : a:Knd fp-> Type
| | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
| | 7. es : ES
| | 8. w : World
| | 9. p : FairFifo
| | 10. PossibleWorld(d-single-effect(i; ds; da; k; x; f);w)
```

```
| | 11. es = ES(w)
| | 12. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| | 13. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| | 14. e : E
| | 15. loc(e) = i
| | 16. kind(e) = k
| | ⊢  (x after e) ∈ ds(x)?Top
| |   by (DoSubsume THEN HypSubst -2 0) THEN BackThruSomeHyp
| |
| .\
| . 1. i : Id
| . 2. x : Id
| . 3. k : Knd
| . 4. ds : x:Id fp-> Type
| . 5. da : a:Knd fp-> Type
| . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
| . 7. es : ES
| . 8. w : World
| . 9. p : FairFifo
| . 10. PossibleWorld(d-single-effect(i; ds; da; k; x; f);w)
| . 11. es = ES(w)
| . 12. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . 13. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . 14. e : E
| . 15. loc(e) = i
| . 16. kind(e) = k
| . ⊢  f((λz.(z when e)),val(e)) ∈ ds(x)?Top  by DoSubsume
| . |
| . |\
| . | .....subterm .....
| . | 1. i : Id
| . | 2. x : Id
| . | 3. k : Knd
| . | 4. ds : x:Id fp-> Type
| . | 5. da : a:Knd fp-> Type
| . | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
| . | 7. es : ES
| . | 8. w : World
| . | 9. p : FairFifo
| . | 10. PossibleWorld(d-single-effect(i; ds; da; k; x; f);w)
| . | 11. es = ES(w)
| . | 12. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . | 13. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . | 14. e : E
| . | 15. loc(e) = i
| . | 16. kind(e) = k
| . | ⊢  (λz.(z when e)) ∈ State(ds)
| . |   by Unfold ‘ma-state‘ 0 THEN DoSubsume THEN HypSubst -3 0
| . |      THEN
| . |      BackThruSomeHyp
| . |
| . .\
| . . .....subterm .....
| . . 1. i : Id
| . . 2. x : Id
```

```
| . . 3. k : Knd
| . . 4. ds : x:Id fp-> Type
| . . 5. da : a:Knd fp-> Type
| . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
| . . 7. es : ES
| . . 8. w : World
| . . 9. p : FairFifo
| . . 10. PossibleWorld(d-single-effect(i; ds; da; k; x; f);w)
| . . 11. es = ES(w)
| . . 12. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
| . . 13. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
| . . 14. e : E
| . . 15. loc(e) = i
| . . 16. kind(e) = k
| . . ⊢  val(e) ∈ ma-valtype(da; k)
| . .   by (DoSubsume THEN RevHypSubst -1 0) THEN BackThruSomeHyp
|
.\
. 1. i : Id
. 2. x : Id
. 3. k : Knd
. 4. ds : x:Id fp-> Type
. 5. da : a:Knd fp-> Type
. 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. ⊢  d-single-effect(i; ds; da; k; x; f
. |                 ) realizes2 es.(∀x:Id. vartype(i;x) ⊆r ds(x)?Top)
. |  & (∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e))))
. |  & (∀e:E.
. |     loc(e) = i
. |     ⇒
. |     kind(e) = k ⇒ (x after e) = f((λz.(z when e)),val(e)))
. | by RepeatFor 3 (Analyze 0 THENA Complete Auto) THEN Analyze 0
. |
. |\
. | 1. i : Id
. | 2. x : Id
. | 3. k : Knd
. | 4. ds : x:Id fp-> Type
. | 5. da : a:Knd fp-> Type
. | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | 7. w : World
. | 8. p : FairFifo
. | 9. PossibleWorld(d-single-effect(i; ds; da; k; x; f);w)
. | ⊢  (∀x:Id. vartype(i;x) ⊆r ds(x)?Top)
. | |  & (∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e))))
. | | by PW_Reduce1 -1 THEN Thin -1
. | |    THEN
. | |    AllHyps (λh.InstHyp [i] h THENA Complete Auto THEN Thin h)
. | |    THEN
. | |    AllHyps
. | |    (λh.
. | |     (RWO
. | |      Thm* ∀d:EqDecider(A), i:A. (eqof(d)(i,i)) ∼ true2
. | |      {eqof_eq_btrue}
. | |      h)
```

```
. | |      THEN
. | |      (Reduce h))
. | |     THEN
. | |     ES_Reduce
. | |     THEN
. | |     Try BackThruSomeHyp
. | |
. | 1. i : Id
. | 2. x : Id
. | 3. k : Knd
. | 4. ds : x:Id fp-> Type
. | 5. da : a:Knd fp-> Type
. | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | 7. w : World
. | 8. p : FairFifo
. | 9. FairFifo
. | 10. ∀l:IdLnk, tg:Id.
. | |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | |                              <ds,da,,,<k,x> : f,,,,·>
. | |                           else  fi))(rcv(l; tg))?Top
. | 11. ∀t:ℕ.
. | |   ¬isnull(a(i;t))
. | |   ⇒
. | |   (islocal(kind(a(i;t)))
. | |    ⇒
. | |    P != (act(kind(a(i;t)))) ==> P((λx.s(i;t).x),val(a(i;t))))
. | |   & (∀x@0:Id.
. | |      E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | |                                             =
. | |                                             E
. | |                                             ((λx.s(i;t).x)
. | |                                             ,val(a(i;t))))
. | |   & (∀l:IdLnk.
. | |      <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | |      s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | |   & (∀x@0:Id.
. | |      ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | |      ⇒
. | |      s(i;t).x@0 = s(i;t+1).x@0)
. | |   & (∀l:IdLnk, tg:Id.
. | |      ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. | |      ⇒
. | |      w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | 13. ∀a:Action(i). ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | 14. ∀x@0:Id. vartype(i;x@0) ⊆r ds(x@0)?Top
. | 15. e : E
. | 16. loc(e) = i
. | ⊢  valtype(e) ⊆r ma-valtype(da; kind(e))  by Thin -3
. | |
. | 1. i : Id
. | 2. x : Id
. | 3. k : Knd
. | 4. ds : x:Id fp-> Type
. | 5. da : a:Knd fp-> Type
```

```
. | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | 7. w : World
. | 8. p : FairFifo
. | 9. FairFifo
. | 10. ∀l:IdLnk, tg:Id.
. | |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | |                             <ds,da,,,<k,x> : f,,,,·>
. | |                          else  fi))(rcv(l; tg))?Top
. | 11. ∀t:ℕ.
. | |   ¬isnull(a(i;t))
. | |   ⇒
. | |   (islocal(kind(a(i;t)))
. | |    ⇒
. | |    P != (act(kind(a(i;t)))) ==> P((λx.s(i;t).x),val(a(i;t))))
. | |   & (∀x@0:Id.
. | |      E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | |                                             =
. | |                                             E
. | |                                             ((λx.s(i;t).x)
. | |                                             ,val(a(i;t))))
. | |   & (∀l:IdLnk.
. | |      <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | |      s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | |   & (∀x@0:Id.
. | |      ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | |      ⇒
. | |      s(i;t).x@0 = s(i;t+1).x@0)
. | |   & (∀l:IdLnk, tg:Id.
. | |      ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. | |      ⇒
. | |      w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | 13. ∀a:Action(i). ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | 14. e : E
. | 15. loc(e) = i
. | ⊢  valtype(e) ⊆r ma-valtype(da; kind(e))  by InstHyp [act(e)] -3
. | |
. | |\
. | | .....wf .....
. | | 1. i : Id
. | | 2. x : Id
. | | 3. k : Knd
. | | 4. ds : x:Id fp-> Type
. | | 5. da : a:Knd fp-> Type
. | | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | | 7. w : World
. | | 8. FairFifo
. | | 9. FairFifo
. | | 10. ∀l:IdLnk, tg:Id.
. | |     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | |                               <ds,da,,,<k,x> : f,,,,·>
. | |                            else  fi))(rcv(l; tg))?Top
. | | 11. ∀t:ℕ.
. | |     ¬isnull(a(i;t))
. | |     ⇒
```

```
. | |     (islocal(kind(a(i;t)))
. | |      ⇒
. | |      P != (act(kind(a(i;t)))) ==> P((λx.s(i;t).x),val(a(i;t))))
. | |     & (∀x@0:Id.
. | |        E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | |                                               =
. | |                                               E
. | |                                               ((λx.s(i;t).x)
. | |                                               ,val(a(i;t))))
. | |     & (∀l:IdLnk.
. | |        <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | |        s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | |     & (∀x@0:Id.
. | |        ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | |        ⇒
. | |        s(i;t).x@0 = s(i;t+1).x@0)
. | |     & (∀l:IdLnk, tg:Id.
. | |        ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. | |        ⇒
. | |        w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | | 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | | 13. ∀a:Action(i).
. | |     ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | | 14. e : E
. | | 15. loc(e) = i
. | | ⊢  act(e) ∈ Action(i)
. | |   by DoSubsume THEN Analyze 0 THEN HypSubst -3 -2

. | |
. | |\
. | | .....antecedent .....
. | | 1. i : Id
. | | 2. x : Id
. | | 3. k : Knd
. | | 4. ds : x:Id fp-> Type
. | | 5. da : a:Knd fp-> Type
. | | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | | 7. w : World
. | | 8. FairFifo
. | | 9. FairFifo
. | | 10. ∀l:IdLnk, tg:Id.
. | |     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | |                                 <ds,da,,,<k,x> : f,,,,·>
. | |                              else  fi))(rcv(l; tg))?Top
. | | 11. ∀t:ℕ.
. | |     ¬isnull(a(i;t))
. | |     ⇒
. | |     (islocal(kind(a(i;t)))
. | |      ⇒
. | |      P != (act(kind(a(i;t)))) ==> P((λx.s(i;t).x),val(a(i;t))))
. | |     & (∀x@0:Id.
. | |        E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | |                                               =
. | |                                               E
. | |                                               ((λx.s(i;t).x)
. | |                                               ,val(a(i;t))))
```

```
. | |     & (∀l:IdLnk.
. | |        <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | |        s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | |     & (∀x@0:Id.
. | |        ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | |        ⇒
. | |        s(i;t).x@0 = s(i;t+1).x@0)
. | |     & (∀l:IdLnk, tg:Id.
. | |        ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. | |        ⇒
. | |        w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | | 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | | 13. ∀a:Action(i).
. | |     ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | | 14. e : E
. | | 15. loc(e) = i
. | | ⊢  ¬isnull(act(e))
. | |   by Inst
. | |      Thm* ∀the_w:World, e:E.
. | |           act(e) ∈ {a:Action(loc(e))| ¬isnull(a) }
. | |      {w_DASH_act_wf}
. | |      [w;e]
. | |      THEN
. | |      Analyze -1
. | |
. | .\
. | . 1. i : Id
. | . 2. x : Id
. | . 3. k : Knd
. | . 4. ds : x:Id fp-> Type
. | . 5. da : a:Knd fp-> Type
. | . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | . 7. w : World
. | . 8. p : FairFifo
. | . 9. FairFifo
. | . 10. ∀l:IdLnk, tg:Id.
. | . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | . |                               <ds,da,,,<k,x> : f,,,,·>
. | . |                           else  fi))(rcv(l; tg))?Top
. | . 11. ∀t:ℕ.
. | . |   ¬isnull(a(i;t))
. | . |   ⇒
. | . |   (islocal(kind(a(i;t)))
. | . |    ⇒
. | . |    P != (act(kind(a(i;t)))) ==> P((λx.s(i;t).x),val(a(i;t))))
. | . |   & (∀x@0:Id.
. | . |      E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | . |                                             =
. | . |                                             E
. | . |                                             ((λx.s(i;t).x)
. | . |                                             ,val(a(i;t))))
. | . |   & (∀l:IdLnk.
. | . |      <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | . |      s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | . |   & (∀x@0:Id.
```

```
. | . |       ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . |       ⇒
. | . |       s(i;t).x@0 = s(i;t+1).x@0)
. | . |    & (∀l:IdLnk, tg:Id.
. | . |       ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . |       ⇒
. | . |       w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | . 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | . 13. ∀a:Action(i).
. | . |    ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | . 14. e : E
. | . 15. loc(e) = i
. | . 16. valtype(i;act(e)) ⊆r da(kind(act(e)))?Top
. | . ⊢  valtype(e) ⊆r ma-valtype(da; kind(e))
. | . | by Fold ‘ma-valtype‘ -1 THEN NthHypEq -1 THEN Analyze
. | . |
. | . |\
. | . | .....subterm .....
. | . | 1. i : Id
. | . | 2. x : Id
. | . | 3. k : Knd
. | . | 4. ds : x:Id fp-> Type
. | . | 5. da : a:Knd fp-> Type
. | . | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | . | 7. w : World
. | . | 8. p : FairFifo
. | . | 9. FairFifo
. | . | 10. ∀l:IdLnk, tg:Id.
. | . | |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | . | |                             <ds,da,,,<k,x> : f,,,,·>
. | . | |                          else  fi))(rcv(l; tg))?Top
. | . | 11. ∀t:ℕ.
. | . | |   ¬isnull(a(i;t))
. | . | |   ⇒
. | . | |   (islocal(kind(a(i;t)))
. | . | |    ⇒
. | . | |    P != (act(kind(a(i;t)))) ==> P
. | . | |                                 ((λx.s(i;t).x)
. | . | |                                 ,val(a(i;t))))
. | . | |   & (∀x@0:Id.
. | . | |      E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | . | |                                             =
. | . | |                                             E
. | . | |                                             ((λx.s(i;t).x)
. | . | |                                             ,val(a(i;t))))
. | . | |   & (∀l:IdLnk.
. | . | |      <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | . | |      s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | . | |   & (∀x@0:Id.
. | . | |      ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . | |      ⇒
. | . | |      s(i;t).x@0 = s(i;t+1).x@0)
. | . | |   & (∀l:IdLnk, tg:Id.
. | . | |      ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . | |      ⇒
```

```
. | . | |      w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | . | 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | . | 13. ∀a:Action(i).
. | . | |   ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | . | 14. e : E
. | . | 15. loc(e) = i
. | . | 16. valtype(i;act(e)) ⊆r ma-valtype(da; kind(act(e)))
. | . | ⊢  valtype(e) = valtype(i;act(e))
. | . | | by Assert (¬isnull(act(e)))
. | . | |
. | . | |\
. | . | | .....assertion .....
. | . | | 1. i : Id
. | . | | 2. x : Id
. | . | | 3. k : Knd
. | . | | 4. ds : x:Id fp-> Type
. | . | | 5. da : a:Knd fp-> Type
. | . | | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | . | | 7. w : World
. | . | | 8. FairFifo
. | . | | 9. FairFifo
. | . | | 10. ∀l:IdLnk, tg:Id.
. | . | |     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | . | |                               <ds,da,,,<k,x> : f,,,,·>
. | . | |                            else  fi))(rcv(l; tg))?Top
. | . | | 11. ∀t:ℕ.
. | . | |     ¬isnull(a(i;t))
. | . | |     ⇒
. | . | |     (islocal(kind(a(i;t)))
. | . | |      ⇒
. | . | |      P != (act(kind(a(i;t)))) ==> P
. | . | |                                   ((λx.s(i;t).x)
. | . | |                                   ,val(a(i;t))))
. | . | |     & (∀x@0:Id.
. | . | |        E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | . | |                                               =
. | . | |                                               E
. | . | |                                               ((λx.s(i;t).x)
. | . | |                                               ,val(a(i;t))))
. | . | |     & (∀l:IdLnk.
. | . | |        <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | . | |        s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | . | |     & (∀x@0:Id.
. | . | |        ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . | |        ⇒
. | . | |        s(i;t).x@0 = s(i;t+1).x@0)
. | . | |     & (∀l:IdLnk, tg:Id.
. | . | |        ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . | |        ⇒
. | . | |        w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | . | | 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | . | | 13. ∀a:Action(i).
. | . | |     ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | . | | 14. e : E
. | . | | 15. loc(e) = i
```

```
. | . | | 16. valtype(i;act(e)) ⊆r ma-valtype(da; kind(act(e)))
. | . | | ⊢  ¬isnull(act(e))
. | . | |   by Inst
. | . | |      Thm* ∀the_w:World, e:E.
. | . | |           act(e) ∈ {a:Action(loc(e))| ¬isnull(a) }
. | . | |      {w_DASH_act_wf}
. | . | |      [w;e]
. | . | |      THEN
. | . | |      Analyze -1
. | . | |
. | . | .\
. | . | . 1. i : Id
. | . | . 2. x : Id
. | . | . 3. k : Knd
. | . | . 4. ds : x:Id fp-> Type
. | . | . 5. da : a:Knd fp-> Type
. | . | . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | . | . 7. w : World
. | . | . 8. p : FairFifo
. | . | . 9. FairFifo
. | . | . 10. ∀l:IdLnk, tg:Id.
. | . | . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | . | . |                              <ds,da,,,<k,x> : f,,,,·>
. | . | . |                           else  fi))(rcv(l; tg))?Top
. | . | . 11. ∀t:ℕ.
. | . | . |   ¬isnull(a(i;t))
. | . | . |   ⇒
. | . | . |   (islocal(kind(a(i;t)))
. | . | . |    ⇒
. | . | . |    P != (act(kind(a(i;t)))) ==> P
. | . | . |                                 ((λx.s(i;t).x)
. | . | . |                                 ,val(a(i;t))))
. | . | . |   & (∀x@0:Id.
. | . | . |      E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | . | . |                                             =
. | . | . |                                             E
. | . | . |                                             ((λx.s(i;t).x)
. | . | . |                                             ,val(a(i;t))))
. | . | . |   & (∀l:IdLnk.
. | . | . |      <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | . | . |      s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | . | . |   & (∀x@0:Id.
. | . | . |      ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . | . |      ⇒
. | . | . |      s(i;t).x@0 = s(i;t+1).x@0)
. | . | . |   & (∀l:IdLnk, tg:Id.
. | . | . |      ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . | . |      ⇒
. | . | . |      w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | . | . 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | . | . 13. ∀a:Action(i).
. | . | . |   ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | . | . 14. e : E
. | . | . 15. loc(e) = i
. | . | . 16. valtype(i;act(e)) ⊆r ma-valtype(da; kind(act(e)))
```

```
. | . | . 17. ¬isnull(act(e))
. | . | . ⊢  valtype(e) = valtype(i;act(e))
. | . | . | by Subst (valtype(e) ~ valtype(loc(e);act(e))) 0
. | . | . |
. | . | . |\
. | . | . | .....equality .....
. | . | . | 1. i : Id
. | . | . | 2. x : Id
. | . | . | 3. k : Knd
. | . | . | 4. ds : x:Id fp-> Type
. | . | . | 5. da : a:Knd fp-> Type
. | . | . | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | . | . | 7. w : World
. | . | . | 8. p : FairFifo
. | . | . | 9. FairFifo
. | . | . | 10. ∀l:IdLnk, tg:Id.
. | . | . |     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | . | . |                                <ds,da,,,<k,x> : f,,,,·>
. | . | . |                             else  fi))(rcv(l; tg))?Top
. | . | . | 11. ∀t:ℕ.
. | . | . |     ¬isnull(a(i;t))
. | . | . |     ⇒
. | . | . |     (islocal(kind(a(i;t)))
. | . | . |      ⇒
. | . | . |      P != (act(kind(a(i;t)))) ==> P
. | . | . |                                    ((λx.s(i;t).x)
. | . | . |                                    ,val(a(i;t))))
. | . | . |     & (∀x@0:Id.
. | . | . |        E != <k,x> : f(<kind(a(i;t))
. | . | . |                       ,x@0>) ==> s(i;t+1).x@0
. | . | . |                                  =
. | . | . |                                  E
. | . | . |                                  ((λx.s(i;t).x)
. | . | . |                                  ,val(a(i;t))))
. | . | . |     & (∀l:IdLnk.
. | . | . |        <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | . | . |        s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | . | . |     & (∀x@0:Id.
. | . | . |        ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . | . |        ⇒
. | . | . |        s(i;t).x@0 = s(i;t+1).x@0)
. | . | . |     & (∀l:IdLnk, tg:Id.
. | . | . |        ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(...);L)
. | . | . |        ⇒
. | . | . |        w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | . | . | 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | . | . | 13. ∀a:Action(i).
. | . | . |     ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | . | . | 14. e : E
. | . | . | 15. loc(e) = i
. | . | . | 16. valtype(i;act(e)) ⊆r ma-valtype(da; kind(act(e)))
. | . | . | 17. ¬isnull(act(e))

. | . | . | ⊢  valtype(e) ~ valtype(loc(e);act(e))
. | . | . |   by Repeat
. | . | . |      (Unfolds
```

```
. | . | . |       [‘w-valtype‘
. | . | . |       ;‘es-valtype‘
. | . | . |       ;‘kindcase‘
. | . | . |       ;‘es-isrcv‘
. | . | . |       ;‘es-kind‘
. | . | . |       ;‘w-ekind‘
. | . | . |       ;‘es-rcvtype‘
. | . | . |       ;‘es-acttype‘
. | . | . |       ;‘w-M‘
. | . | . |       ;‘w-V‘
. | . | . |       ;‘es-lnk‘
. | . | . |       ;‘es-tag‘
. | . | . |       ;‘es-loc‘
. | . | . |       ;‘es-act‘]
. | . | . |       0
. | . | . |       THEN
. | . | . |       Reduce 0)
. | . | . |      THEN
. | . | . |      SplitOnConclITE
. | . | . |      THEN
. | . | . |      SplitOnConclITE
. | . | . |      THEN
. | . | . |      Try
. | . | . |      (RWO
. | . | . |       Thm* ∀k:Knd. islocal(k) ~ ¬₂isrcv(k)
. | . | . |       {islocal_DASH_not_DASH_isrcv}
. | . | . |       -1
. | . | . |       THEN
. | . | . |       RW assert_pushdownC -1
. | . | . |       THEN
. | . | . |       Complete Auto)
. | . | . |      THEN
. | . | . |      Try (Fold ‘w-TA‘ 0 THEN Trivial)
. | . | . |
. | . | . .\
. | . | . . 1. i : Id
. | . | . . 2. x : Id
. | . | . . 3. k : Knd
. | . | . . 4. ds : x:Id fp-> Type
. | . | . . 5. da : a:Knd fp-> Type
. | . | . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | . | . . 7. w : World
. | . | . . 8. FairFifo
. | . | . . 9. FairFifo
. | . | . . 10. ∀l:IdLnk, tg:Id.
. | . | . .     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | . | . .                               <ds,da,,,<k,x> : f,,,,⋅>
. | . | . .                            else  fi))(rcv(l; tg))?Top
. | . | . . 11. ∀t:ℕ.
. | . | . .     ¬isnull(a(i;t))
. | . | . .     ⇒
. | . | . .     (islocal(kind(a(i;t)))
. | . | . .      ⇒
. | . | . .      P != (act(kind(a(i;t)))) ==> P
. | . | . .                                 ((λx.s(i;t).x)
```

```
. | . | . .                                   ,val(a(i;t))))
. | . | . .     & (∀x@0:Id.
. | . | . .        E != <k,x> : f(<kind(a(i;t))
. | . | . .                        ,x@0>) ==> s(i;t+1).x@0
. | . | . .                                   =
. | . | . .                                   E
. | . | . .                                   ((λx.s(i;t).x)
. | . | . .                                   ,val(a(i;t))))
. | . | . .     & (∀l:IdLnk.
. | . | . .        <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | . | . .        s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | . | . .     & (∀x@0:Id.
. | . | . .        ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . | . .        ⇒
. | . | . .        s(i;t).x@0 = s(i;t+1).x@0)
. | . | . .     & (∀l:IdLnk, tg:Id.
. | . | . .        ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(...);L)
. | . | . .        ⇒
. | . | . .        w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | . | . . 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | . | . . 13. ∀a:Action(i).
. | . | . .     ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | . | . . 14. e : E
. | . | . . 15. loc(e) = i
. | . | . . 16. valtype(i;act(e)) ⊆r ma-valtype(da; kind(act(e)))
. | . | . . 17. ¬isnull(act(e))
. | . | . . ⊢  valtype(loc(e);act(e)) = valtype(i;act(e))  by Analyze
. | . |
. | . .\
. | . . .....subterm .....
. | . . 1. i : Id
. | . . 2. x : Id
. | . . 3. k : Knd
. | . . 4. ds : x:Id fp-> Type
. | . . 5. da : a:Knd fp-> Type
. | . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | . . 7. w : World
. | . . 8. FairFifo
. | . . 9. FairFifo
. | . . 10. ∀l:IdLnk, tg:Id.
. | . . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | . . |                              <ds,da,,,<k,x> : f,,,,·>

. | . . |                           else  fi))(rcv(l; tg))?Top
. | . . 11. ∀t:ℕ.
. | . . |   ¬isnull(a(i;t))
. | . . |   ⇒
. | . . |   (islocal(kind(a(i;t)))
. | . . |    ⇒
. | . . |    P != (act(kind(a(i;t)))) ==> P
. | . . |                                 ((λx.s(i;t).x)
. | . . |                                 ,val(a(i;t))))
. | . . |   & (∀x@0:Id.
. | . . |      E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | . . |                                             =
. | . . |                                             E
```

```
. | . . |                                                 ((λx.s(i;t).x)
. | . . |                                                 ,val(a(i;t))))
. | . . |   & (∀l:IdLnk.
. | . . |      <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | . . |      s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | . . |   & (∀x@0:Id.
. | . . |      ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . . |      ⇒
. | . . |      s(i;t).x@0 = s(i;t+1).x@0)
. | . . |   & (∀l:IdLnk, tg:Id.
. | . . |      ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . . |      ⇒
. | . . |      w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | . . 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | . . 13. ∀a:Action(i).
. | . . |   ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | . . 14. e : E
. | . . 15. loc(e) = i
. | . . 16. valtype(i;act(e)) ⊆r ma-valtype(da; kind(act(e)))
. | . . ⊢  ma-valtype(da; kind(e)) = ma-valtype(da; kind(act(e)))
. | . . | by Unfold ‘w-ekind‘ 0
. | . . |
. | . . 1. i : Id
. | . . 2. x : Id
. | . . 3. k : Knd
. | . . 4. ds : x:Id fp-> Type
. | . . 5. da : a:Knd fp-> Type
. | . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | . . 7. w : World
. | . . 8. FairFifo
. | . . 9. FairFifo
. | . . 10. ∀l:IdLnk, tg:Id.
. | . . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | . . |                             <ds,da,,,<k,x> : f,,,,·>
. | . . |                          else  fi))(rcv(l; tg))?Top
. | . . 11. ∀t:ℕ.
. | . . |   ¬isnull(a(i;t))
. | . . |   ⇒
. | . . |   (islocal(kind(a(i;t)))
. | . . |    ⇒
. | . . |    P != (act(kind(a(i;t)))) ==> P
. | . . |                                 ((λx.s(i;t).x)
. | . . |                                 ,val(a(i;t))))
. | . . |   & (∀x@0:Id.
. | . . |      E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | . . |                                             =
. | . . |                                             E
. | . . |                                             ((λx.s(i;t).x)
. | . . |                                             ,val(a(i;t))))
. | . . |   & (∀l:IdLnk.
. | . . |      <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | . . |      s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | . . |   & (∀x@0:Id.
. | . . |      ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . . |      ⇒
```

```
. | . . |       s(i;t).x@0 = s(i;t+1).x@0)
. | . . |    & (∀l:IdLnk, tg:Id.
. | . . |       ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . . |       ⇒
. | . . |       w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | . . 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | . . 13. ∀a:Action(i).
. | . . |    ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | . . 14. e : E
. | . . 15. loc(e) = i
. | . . 16. valtype(i;act(e)) ⊆r ma-valtype(da; kind(act(e)))
. | . . ⊢  ma-valtype(da; kind(act(e))) = ma-valtype(da; kind(act(e)))
. | . . | by Subst
. | . . |    (ma-valtype(da; kind(act(e))) ~ ma-valtype(da;
. | . . |                                                  kind(act(e))))
. | . . |    0
. | . . |
. | . . |\
. | . . | .....equality .....
. | . . | 1. i : Id
. | . . | 2. x : Id
. | . . | 3. k : Knd
. | . . | 4. ds : x:Id fp-> Type
. | . . | 5. da : a:Knd fp-> Type
. | . . | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | . . | 7. w : World
. | . . | 8. FairFifo
. | . . | 9. FairFifo
. | . . | 10. ∀l:IdLnk, tg:Id.
. | . . |     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | . . |                               <ds,da,,,<k,x> : f,,,,·>
. | . . |                            else  fi))(rcv(l; tg))?Top
. | . . | 11. ∀t:ℕ.
. | . . |     ¬isnull(a(i;t))
. | . . |     ⇒
. | . . |     (islocal(kind(a(i;t)))
. | . . |      ⇒
. | . . |      P != (act(kind(a(i;t)))) ==> P
. | . . |                                   ((λx.s(i;t).x)
. | . . |                                   ,val(a(i;t))))
. | . . |     & (∀x@0:Id.
. | . . |        E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | . . |                                               =
. | . . |                                               E
. | . . |                                               ((λx.s(i;t).x)
. | . . |                                               ,val(a(i;t))))
. | . . |     & (∀l:IdLnk.
. | . . |        <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | . . |        s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | . . |     & (∀x@0:Id.
. | . . |        ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . . |        ⇒
. | . . |        s(i;t).x@0 = s(i;t+1).x@0)
. | . . |     & (∀l:IdLnk, tg:Id.
. | . . |        ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
```

```
. | . . |        ⇒
. | . . |        w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | . . | 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | . . | 13. ∀a:Action(i).
. | . . |     ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | . . | 14. e : E
. | . . | 15. loc(e) = i
. | . . | 16. valtype(i;act(e)) ⊆r ma-valtype(da; kind(act(e)))
. | . . | ⊢  ma-valtype(da; kind(act(e))) ~ ma-valtype(da;
. | . . |                                         kind(act(e)))
. | . . |   by Unfold `ma-valtype` 0 THEN PWTrivial
. | . . |
. | . . .\
. | . . . 1. i : Id
. | . . . 2. x : Id
. | . . . 3. k : Knd
. | . . . 4. ds : x:Id fp-> Type
. | . . . 5. da : a:Knd fp-> Type
. | . . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | . . . 7. w : World
. | . . . 8. FairFifo
. | . . . 9. FairFifo
. | . . . 10. ∀l:IdLnk, tg:Id.
. | . . . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | . . . |                          <ds,da,,,<k,x> : f,,,,·>
. | . . . |                       else  fi))(rcv(l; tg))?Top
. | . . . 11. ∀t:ℕ.
. | . . . |   ¬isnull(a(i;t))
. | . . . |   ⇒
. | . . . |   (islocal(kind(a(i;t)))
. | . . . |    ⇒
. | . . . |    P != (act(kind(a(i;t)))) ==> P
. | . . . |                               ((λx.s(i;t).x)
. | . . . |                               ,val(a(i;t))))
. | . . . |   & (∀x@0:Id.
. | . . . |      E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | . . . |                                           =
. | . . . |                                           E
. | . . . |                                           ((λx.s(i;t).x)
. | . . . |                                           ,val(a(i;t))))
. | . . . |   & (∀l:IdLnk.
. | . . . |      <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | . . . |      s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | . . . |   & (∀x@0:Id.
. | . . . |      ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . . . |      ⇒
. | . . . |      s(i;t).x@0 = s(i;t+1).x@0)
. | . . . |   & (∀l:IdLnk, tg:Id.
. | . . . |      ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . . . |      ⇒
. | . . . |      w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | . . . 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | . . . 13. ∀a:Action(i).
. | . . . |   ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | . . . 14. e : E
```

```
. | . . . 15. loc(e) = i
. | . . . 16. valtype(i;act(e)) ⊆r ma-valtype(da; kind(act(e)))
. | . . . ⊢  ma-valtype(da; kind(act(e)))
. | . . . |  =
. | . . . |  ma-valtype(da; kind(act(e)))  by Auto
. | . . . |
. | . . . 1. i : Id
. | . . . 2. x : Id
. | . . . 3. k : Knd
. | . . . 4. ds : x:Id fp-> Type
. | . . . 5. da : a:Knd fp-> Type
. | . . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. | . . . 7. w : World
. | . . . 8. FairFifo
. | . . . 9. FairFifo
. | . . . 10. ∀l:IdLnk, tg:Id.
. | . . .     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. | . . .                               <ds,da,,,<k,x> : f,,,,·>
. | . . .                            else  fi))(rcv(l; tg))?Top
. | . . . 11. ∀t:ℕ.
. | . . .     ¬isnull(a(i;t))
. | . . .     ⇒
. | . . .     (islocal(kind(a(i;t)))
. | . . .      ⇒
. | . . .      P != (act(kind(a(i;t)))) ==> P
. | . . .                                   ((λx.s(i;t).x)
. | . . .                                   ,val(a(i;t))))
. | . . .     & (∀x@0:Id.
. | . . .        E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. | . . .                                               =
. | . . .                                               E
. | . . .                                               ((λx.s(i;t).x)
. | . . .                                               ,val(a(i;t))))
. | . . .     & (∀l:IdLnk.
. | . . .        <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. | . . .        s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i))
. | . . .     & (∀x@0:Id.
. | . . .        ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . . .        ⇒
. | . . .        s(i;t).x@0 = s(i;t+1).x@0)
. | . . .     & (∀l:IdLnk, tg:Id.
. | . . .        ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. | . . .        ⇒
. | . . .        w-tagged(tg; onlnk(l;m(i;t))) = nil)
. | . . . 12. ∀x@0:Id. x0 != (x@0) ==> s(i;0).x@0 = x0
. | . . . 13. ∀a:Action(i).
. | . . .     ¬isnull(a) ⇒ (valtype(i;a) ⊆r da(kind(a))?Top)
. | . . . 14. e : E
. | . . . 15. loc(e) = i
. | . . . 16. valtype(i;act(e)) ⊆r ma-valtype(da; kind(act(e)))
. | . . . ⊢  ¬isnull(act(e))  by Unfold ‘w-act‘ 0 THEN DVar ‘e’
. |
. .\
. . 1. i : Id
. . 2. x : Id
```

```
. . 3. k : Knd
. . 4. ds : x:Id fp-> Type
. . 5. da : a:Knd fp-> Type
. . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . 7. w : World
. . 8. p : FairFifo
. . 9. PossibleWorld(d-single-effect(i; ds; da; k; x; f);w)
. . 10. (∀x:Id. vartype(i;x) ⊆r ds(x)?Top)
. . |   & (∀e:E.
. . |      loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e))))
. . ⊢  ∀e:E.
. . |  loc(e) = i
. . |  ⇒
. . |  kind(e) = k ⇒ (x after e) = f((λz.(z when e)),val(e))
. . | by ES_Reduce THEN PW_Reduce1 -2
. . |
. . 1. i : Id
. . 2. x : Id
. . 3. k : Knd
. . 4. ds : x:Id fp-> Type
. . 5. da : a:Knd fp-> Type
. . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . 7. w : World
. . 8. p : FairFifo
. . 9. FairFifo
. . 10. ∀i@0,x@0:Id.
. . |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . |                              <ds,da,,,<k,x> : f,,,,·>
. . |                           else  fi)(x@0)?Top
. . 11. ∀i@0:Id, a:Action(i@0).
. . |   ¬isnull(a)
. . |   ⇒
. . |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . |                                 <ds,da,,,<k,x> : f,,,,·>
. . |                              else  fi))(kind(a))?Top)
. . 12. ∀l:IdLnk, tg:Id.
. . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . |                             <ds,da,,,<k,x> : f,,,,·>
. . |                          else  fi))(rcv(l; tg))?Top
. . 13. ∀i@0,x@0:Id.
. . |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . |                        <ds,da,,,<k,x> : f,,,,·>
. . |                     else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . 14. ∀i@0:Id, t:ℕ.
. . |   ¬isnull(a(i@0;t))
. . |   ⇒
. . |   (islocal(kind(a(i@0;t)))
. . |    ⇒
. . |    P != 1of(
. . |         2of(2of(2of(
. . |         if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . |         else  fi))))(act(kind(a(i@0;t)))) ==> P
. . |                                            ((λx.s(i@0;t).x)
. . |                                            ,val(a(i@0;t))))
. . |   & (∀x@0:Id.
```

```
. . |      E != 1of(
. . |           2of(2of(2of(2of(
. . |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . |           else  fi)))))(<kind(a(i@0;t))
. . |                         ,x@0>) ==> s(i@0;t+1).x@0
. . |                                    =
. . |                                    E
. . |                                    ((λx.s(i@0;t).x)
. . |                                    ,val(a(i@0;t))))
. . |   & (∀l:IdLnk.
. . |      if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . |      else  fi.send(kind(a(i@0;t));l;λx.

. . |      s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . |   & (∀x@0:Id.
. . |      ¬L != 1of(
. . |            2of(2of(2of(2of(2of(2of(
. . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . |            else  fi)))))))(x@0) ==> deq-member(KindDeq;...;L)
. . |      ⇒
. . |      s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . |   & (∀l:IdLnk, tg:Id.
. . |      ¬L != 1of(
. . |            2of(2of(2of(2of(2of(2of(2of(
. . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . |            else  fi))))))))(<l,tg>) ==> deq-member(...;...;L)
. . |      ⇒
. . |      w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . 15. ∀i@0,a:Id, t:ℕ.
. . |   ∃t':ℕ.
. . |   t≤t'
. . |   & ¬isnull(a(i@0;t')) & kind(a(i@0;t')) = locl(a)
. . |     ∨ ¬a declared in if eqof(IdDeq)(i@0,i)→
. . |                         <ds,da,,,<k,x> : f,,,,·>
. . |                      else  fi
. . |     ∨ unsolvable if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . |                  else  fi.pre(a,λx.s(i@0;t').x)
. . 16. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . 17. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . 18. e : E
. . 19. loc(e) = i
. . 20. kind(e) = k
. . ⊢  (x after e) = f((λz.(z when e)),val(e))
. . | by Thin -6 THEN Analyze -3 THEN Analyze -4 THEN All Reduce
. . |    THEN
. . |    RenameVar `j' -5
. . |    THEN
. . |    RenameVar `t' -4
. . |    THEN
. . |    Decide isnull(a(j;t))
. . |
. . |\
. . | 1. i : Id
. . | 2. x : Id
. . | 3. k : Knd
. . | 4. ds : x:Id fp-> Type
```

```
. . | 5. da : a:Knd fp-> Type
. . | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . | 7. w : World
. . | 8. p : FairFifo
. . | 9. FairFifo
. . | 10. ∀i@0,x@0:Id.
. . |     vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . |                                <ds,da,,,<k,x> : f,,,,·>
. . |                             else  fi)(x@0)?Top
. . | 11. ∀i@0:Id, a:Action(i@0).
. . |     ¬isnull(a)
. . |     ⇒
. . |     (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . |                                   <ds,da,,,<k,x> : f,,,,·>
. . |                                else  fi))(kind(a))?Top)
. . | 12. ∀l:IdLnk, tg:Id.
. . |     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . |                               <ds,da,,,<k,x> : f,,,,·>
. . |                            else  fi))(rcv(l; tg))?Top
. . | 13. ∀i@0,x@0:Id.
. . |     x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . |                          <ds,da,,,<k,x> : f,,,,·>
. . |                       else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . | 14. ∀i@0:Id, t:ℕ.
. . |     ¬isnull(a(i@0;t))
. . |     ⇒
. . |     (islocal(kind(a(i@0;t)))
. . |      ⇒
. . |      P != 1of(
. . |           2of(2of(2of(
. . |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . |           else  fi))))(act(kind(a(i@0;t)))) ==> P
. . |                                                  ((λx.s(i@0;t).x)
. . |                                                  ,val(a(i@0;t))))
. . |     & (∀x@0:Id.
. . |        E != 1of(
. . |             2of(2of(2of(2of(
. . |             if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . |             else  fi)))))(<kind(a(i@0;t))
. . |                           ,x@0>) ==> s(i@0;t+1).x@0
. . |                                      =
. . |                                      E
. . |                                      ((λx.s(i@0;t).x)
. . |                                      ,val(a(i@0;t))))
. . |     & (∀l:IdLnk.
. . |        if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . |        else  fi.send(kind(a(i@0;t));l;λx.
. . |        s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . |     & (∀x@0:Id.
. . |        ¬L != 1of(
. . |              2of(2of(2of(2of(2of(2of(
. . |              if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . |              else  fi)))))))(x@0) ==> deq-member(KindDeq;...;L)
. . |        ⇒
. . |        s(i@0;t).x@0 = s(i@0;t+1).x@0)
```

```
. . |     & (∀l:IdLnk, tg:Id.
. . |        ¬L != 1of(

. . |              2of(2of(2of(2of(2of(2of(2of(
. . |              if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . |              else  fi))))))))(<l,tg>) ==> deq-member(...;...;L)
. . |        ⇒
. . |        w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . | 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . | 16. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . | 17. j : Id
. . | 18. t : ℕ
. . | 19. ¬isnull(a(j;t))
. . | 20. j = i
. . | 21. kind(<j,t>) = k
. . | 22. isnull(a(j;t))
. . | ⊢  (x after <j,t>) = f((λz.(z when <j,t>)),val(<j,t>))
. . |   by Assert False THEN Unhide
. . |
. . .\
. . . 1. i : Id
. . . 2. x : Id
. . . 3. k : Knd
. . . 4. ds : x:Id fp-> Type
. . . 5. da : a:Knd fp-> Type
. . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . 7. w : World
. . . 8. p : FairFifo
. . . 9. FairFifo
. . . 10. ∀i@0,x@0:Id.
. . . |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . |                              <ds,da,,,<k,x> : f,,,,·>
. . . |                           else  fi)(x@0)?Top
. . . 11. ∀i@0:Id, a:Action(i@0).
. . . |   ¬isnull(a)
. . . |   ⇒
. . . |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . |                                 <ds,da,,,<k,x> : f,,,,·>
. . . |                              else  fi))(kind(a))?Top)
. . . 12. ∀l:IdLnk, tg:Id.
. . . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . |                             <ds,da,,,<k,x> : f,,,,·>
. . . |                          else  fi))(rcv(l; tg))?Top
. . . 13. ∀i@0,x@0:Id.
. . . |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . |                        <ds,da,,,<k,x> : f,,,,·>
. . . |                     else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . . 14. ∀i@0:Id, t:ℕ.
. . . |   ¬isnull(a(i@0;t))
. . . |   ⇒
. . . |   (islocal(kind(a(i@0;t)))
. . . |    ⇒
. . . |    P != 1of(
. . . |        2of(2of(2of(
. . . |        if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . |        else  fi))))(act(kind(a(i@0;t)))) ==> P
```

```
. . . |                                               ((λx.s(i@0;t).x)
. . . |                                               ,val(a(i@0;t))))
. . . |   & (∀x@0:Id.
. . . |      E != 1of(
. . . |           2of(2of(2of(2of(
. . . |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . |           else  fi)))))(<kind(a(i@0;t))
. . . |                         ,x@0>) ==> s(i@0;t+1).x@0
. . . |                                    =
. . . |                                    E
. . . |                                    ((λx.s(i@0;t).x)
. . . |                                    ,val(a(i@0;t))))
. . . |   & (∀l:IdLnk.
. . . |      if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . |      else  fi.send(kind(a(i@0;t));l;λx.
. . . |      s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . |   & (∀x@0:Id.
. . . |      ¬L != 1of(
. . . |            2of(2of(2of(2of(2of(2of(
. . . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . |            else  fi)))))))(x@0) ==> deq-member(KindDeq;...;L)
. . . |      ⇒
. . . |      s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . |   & (∀l:IdLnk, tg:Id.
. . . |      ¬L != 1of(
. . . |            2of(2of(2of(2of(2of(2of(2of(
. . . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . |            else  fi))))))))(<l,tg>) ==> deq-member(...;...;L)
. . . |      ⇒
. . . |      w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . 16. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . 17. j : Id
. . . 18. t : ℕ
. . . 19. ¬isnull(a(j;t))
. . . 20. j = i
. . . 21. kind(<j,t>) = k
. . . 22. ¬isnull(a(j;t))
. . . ⊢  (x after <j,t>) = f((λz.(z when <j,t>)),val(<j,t>))
. . . | by Thin -4 THEN HypSubst -3 -1
. . . |
. . . 1. i : Id
. . . 2. x : Id
. . . 3. k : Knd
. . . 4. ds : x:Id fp-> Type
. . . 5. da : a:Knd fp-> Type
. . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void

. . . 7. w : World
. . . 8. p : FairFifo
. . . 9. FairFifo
. . . 10. ∀i@0,x@0:Id.
. . . |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . |                                <ds,da,,,<k,x> : f,,,,·>
. . . |                             else  fi)(x@0)?Top
. . . 11. ∀i@0:Id, a:Action(i@0).
```

```
. . . |   ¬isnull(a)
. . . |   ⇒
. . . |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . |                                 <ds,da,,,<k,x> : f,,,,·>
. . . |                              else  fi))(kind(a))?Top)
. . . 12. ∀l:IdLnk, tg:Id.
. . . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . |                             <ds,da,,,<k,x> : f,,,,·>
. . . |                          else  fi))(rcv(l; tg))?Top
. . . 13. ∀i@0,x@0:Id.
. . . |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . |                        <ds,da,,,<k,x> : f,,,,·>
. . . |                     else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . . 14. ∀i@0:Id, t:ℕ.
. . . |   ¬isnull(a(i@0;t))
. . . |   ⇒
. . . |   (islocal(kind(a(i@0;t)))
. . . |    ⇒
. . . |    P != 1of(
. . . |         2of(2of(2of(
. . . |         if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . |         else  fi))))(act(kind(a(i@0;t)))) ==> P
. . . |                                                ((λx.s(i@0;t).x)
. . . |                                                ,val(a(i@0;t))))
. . . |   & (∀x@0:Id.
. . . |      E != 1of(
. . . |           2of(2of(2of(2of(
. . . |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . |           else  fi)))))(<kind(a(i@0;t))
. . . |                         ,x@0>) ==> s(i@0;t+1).x@0
. . . |                                    =
. . . |                                    E
. . . |                                    ((λx.s(i@0;t).x)
. . . |                                    ,val(a(i@0;t))))
. . . |   & (∀l:IdLnk.
. . . |      if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . |      else  fi.send(kind(a(i@0;t));l;λx.
. . . |      s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . |   & (∀x@0:Id.
. . . |      ¬L != 1of(
. . . |            2of(2of(2of(2of(2of(2of(
. . . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . |            else  fi)))))))(x@0) ==> deq-member(KindDeq;...;L)
. . . |      ⇒
. . . |      s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . |   & (∀l:IdLnk, tg:Id.
. . . |      ¬L != 1of(
. . . |            2of(2of(2of(2of(2of(2of(2of(
. . . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . |            else  fi))))))))(<l,tg>) ==> deq-member(...;...;L)
. . . |      ⇒
. . . |      w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . 16. ∀e:E. loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . 17. j : Id
```

```
. . . 18. t : ℕ
. . . 19. j = i
. . . 20. kind(<j,t>) = k
. . . 21. ¬isnull(a(i;t))
. . . ⊢  (x after <j,t>) = f((λz.(z when <j,t>)),val(<j,t>))
. . . | by Assert (valtype(i;a(i;t)) ⊆r ma-valtype(da; k))
. . . |
. . . |\
. . . | .....assertion .....
. . . | 1. i : Id
. . . | 2. x : Id
. . . | 3. k : Knd
. . . | 4. ds : x:Id fp-> Type
. . . | 5. da : a:Knd fp-> Type
. . . | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . | 7. w : World
. . . | 8. p : FairFifo
. . . | 9. FairFifo
. . . | 10. ∀i@0,x@0:Id.
. . . | |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . | |                              <ds,da,,,<k,x> : f,,,,·>
. . . | |                           else  fi)(x@0)?Top
. . . | 11. ∀i@0:Id, a:Action(i@0).
. . . | |   ¬isnull(a)
. . . | |   ⇒
. . . | |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . | |                                 <ds,da,,,<k,x> : f,,,,·>
. . . | |                              else  fi))(kind(a))?Top)
. . . | 12. ∀l:IdLnk, tg:Id.
. . . | |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . | |                             <ds,da,,,<k,x> : f,,,,·>
. . . | |                          else  fi))(rcv(l; tg))?Top
. . . | 13. ∀i@0,x@0:Id.
. . . | |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . | |                       <ds,da,,,<k,x> : f,,,,·>
. . . | |                    else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . . | 14. ∀i@0:Id, t:ℕ.
. . . | |   ¬isnull(a(i@0;t))
. . . | |   ⇒
. . . | |   (islocal(kind(a(i@0;t)))
. . . | |    ⇒
. . . | |    P != 1of(
. . . | |         2of(2of(2of(
. . . | |         if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |         else  fi))))(act(kind(a(i@0;t)))) ==> ...)
. . . | |   & (∀x@0:Id.
. . . | |      E != 1of(
. . . | |           2of(2of(2of(2of(
. . . | |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |           else  fi)))))(<kind(a(i@0;t))
. . . | |                          ,x@0>) ==> s(i@0;t+1).x@0
. . . | |                                     =
. . . | |                                     E
. . . | |                                     ((λx.s(i@0;t).x)
. . . | |                                     ,val(a(i@0;t))))
```

```
. . . | |    & (∀l:IdLnk.
. . . | |       if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |       else  fi.send(kind(a(i@0;t));l;λx.
. . . | |       s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . | |    & (∀x@0:Id.
. . . | |       ¬L != 1of(
. . . | |             2of(2of(2of(2of(2of(2of(
. . . | |             if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |             else  fi)))))))(x@0) ==> deq-member(...;...;L)
. . . | |       ⇒
. . . | |       s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . | |    & (∀l:IdLnk, tg:Id.
. . . | |       ¬L != 1of(
. . . | |             2of(2of(2of(2of(2of(2of(2of(
. . . | |             if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |             else  fi))))))))(<l,tg>) ==> ...
. . . | |       ⇒
. . . | |       w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . | 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . | 16. ∀e:E.
. . . | |    loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . | 17. j : Id
. . . | 18. t : ℕ
. . . | 19. j = i
. . . | 20. kind(<j,t>) = k
. . . | 21. ¬isnull(a(i;t))
. . . | ⊢  valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . | | by ExRepD THEN InstHyp [<i,t>] -6
. . . | |
. . . | |\
. . . | | .....wf .....
. . . | | 1. i : Id
. . . | | 2. x : Id
. . . | | 3. k : Knd
. . . | | 4. ds : x:Id fp-> Type
. . . | | 5. da : a:Knd fp-> Type
. . . | | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . | | 7. w : World
. . . | | 8. p : FairFifo
. . . | | 9. FairFifo
. . . | | 10. ∀i@0,x@0:Id.
. . . | |     vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . | |                                <ds,da,,,<k,x> : f,,,,·>
. . . | |                             else  fi)(x@0)?Top
. . . | | 11. ∀i@0:Id, a:Action(i@0).
. . . | |     ¬isnull(a)
. . . | |     ⇒
. . . | |     (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . | |                                   <ds,da,,,<k,x> : f,,,,·>
. . . | |                                else  fi))(kind(a))?Top)
. . . | | 12. ∀l:IdLnk, tg:Id.
. . . | |     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . | |                               <ds,da,,,<k,x> : f,,,,·>
. . . | |                            else  fi))(rcv(l; tg))?Top
. . . | | 13. ∀i@0,x@0:Id.
```

```
. . . | |     x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . | |                           <ds,da,,,<k,x> : f,,,,·>
. . . | |                       else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . . | | 14. ∀i@0:Id, t:ℕ.
. . . | |     ¬isnull(a(i@0;t))
. . . | |     ⇒
. . . | |     (islocal(kind(a(i@0;t)))
. . . | |      ⇒
. . . | |      P != 1of(
. . . | |           2of(2of(2of(
. . . | |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |           else  fi))))(act(kind(a(i@0;t)))) ==> P
. . . | |                                                (...
. . . | |                                                ,val(...)))
. . . | |     & (∀x@0:Id.
. . . | |        E != 1of(
. . . | |             2of(2of(2of(2of(
. . . | |             if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |             else  fi)))))(<kind(a(i@0;t))
. . . | |                           ,x@0>) ==> s(i@0;t+1).x@0
. . . | |                                      =
. . . | |                                      E
. . . | |                                      ((λx.s(i@0;t).x)
. . . | |                                      ,val(a(i@0;t))))
. . . | |     & (∀l:IdLnk.
. . . | |        if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |        else  fi.send(kind(a(i@0;t));l;λx.

. . . | |        s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . | |     & (∀x@0:Id.
. . . | |        ¬L != 1of(
. . . | |              2of(2of(2of(2of(2of(2of(
. . . | |              if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |              else  fi)))))))(x@0) ==> deq-member(...;...;L)
. . . | |        ⇒
. . . | |        s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . | |     & (∀l:IdLnk, tg:Id.
. . . | |        ¬L != 1of(
. . . | |              2of(2of(2of(2of(2of(2of(2of(
. . . | |              if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |              else  fi))))))))(<l,tg>) ==> ...
. . . | |        ⇒
. . . | |        w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . | | 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . | | 16. ∀e:E.
. . . | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . | | 17. j : Id
. . . | | 18. t : ℕ
. . . | | 19. j = i
. . . | | 20. kind(<j,t>) = k
. . . | | 21. ¬isnull(a(i;t))
. . . | | ⊢  <i,t> ∈ E
. . . | |   by ES_Reduce THEN Unfold ‘w-E‘ 0 THEN Analyze
. . . | |      THEN
. . . | |      Reduce 0
. . . | |
```

```
. . . | |\
. . . | | .....antecedent .....
. . . | | 1. i : Id
. . . | | 2. x : Id
. . . | | 3. k : Knd
. . . | | 4. ds : x:Id fp-> Type
. . . | | 5. da : a:Knd fp-> Type
. . . | | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . | | 7. w : World
. . . | | 8. p : FairFifo
. . . | | 9. FairFifo
. . . | | 10. ∀i@0,x@0:Id.
. . . | |     vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . | |                                <ds,da,,,<k,x> : f,,,,·>
. . . | |                             else  fi)(x@0)?Top
. . . | | 11. ∀i@0:Id, a:Action(i@0).
. . . | |     ¬isnull(a)
. . . | |     ⇒
. . . | |     (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . | |                                   <ds,da,,,<k,x> : f,,,,·>
. . . | |                                else  fi))(kind(a))?Top)
. . . | | 12. ∀l:IdLnk, tg:Id.
. . . | |     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . | |                               <ds,da,,,<k,x> : f,,,,·>
. . . | |                            else  fi))(rcv(l; tg))?Top
. . . | | 13. ∀i@0,x@0:Id.
. . . | |     x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . | |                          <ds,da,,,<k,x> : f,,,,·>
. . . | |                       else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . . | | 14. ∀i@0:Id, t:ℕ.
. . . | |     ¬isnull(a(i@0;t))
. . . | |     ⇒
. . . | |     (islocal(kind(a(i@0;t)))
. . . | |      ⇒
. . . | |      P != 1of(
. . . | |           2of(2of(2of(
. . . | |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |           else  fi))))(act(kind(a(i@0;t)))) ==> P
. . . | |                                                 (...
. . . | |                                                 ,val(...)))
. . . | |     & (∀x@0:Id.
. . . | |        E != 1of(
. . . | |             2of(2of(2of(2of(
. . . | |             if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |             else  fi)))))(<kind(a(i@0;t))
. . . | |                           ,x@0>) ==> s(i@0;t+1).x@0
. . . | |                                      =
. . . | |                                      E
. . . | |                                      ((λx.s(i@0;t).x)
. . . | |                                      ,val(a(i@0;t))))
. . . | |     & (∀l:IdLnk.
. . . | |        if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |        else  fi.send(kind(a(i@0;t));l;λx.
. . . | |        s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . | |     & (∀x@0:Id.
```

```
. . . | |        ¬L != 1of(
. . . | |              2of(2of(2of(2of(2of(2of(
. . . | |              if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |              else  fi)))))))(x@0) ==> deq-member(...;...;L)
. . . | |        ⇒
. . . | |        s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . | |     & (∀l:IdLnk, tg:Id.
. . . | |        ¬L != 1of(
. . . | |              2of(2of(2of(2of(2of(2of(2of(
. . . | |              if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | |              else  fi))))))))(<l,tg>) ==> ...
. . . | |        ⇒
. . . | |        w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . | | 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . | | 16. ∀e:E.
. . . | |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . | | 17. j : Id
. . . | | 18. t : ℕ
. . . | | 19. j = i
. . . | | 20. kind(<j,t>) = k
. . . | | 21. ¬isnull(a(i;t))
. . . | | ⊢  loc(<i,t>) = i  by ES_Reduce
. . . | |
. . . | .\
. . . | . 1. i : Id
. . . | . 2. x : Id
. . . | . 3. k : Knd
. . . | . 4. ds : x:Id fp-> Type
. . . | . 5. da : a:Knd fp-> Type
. . . | . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . | . 7. w : World
. . . | . 8. p : FairFifo
. . . | . 9. FairFifo
. . . | . 10. ∀i@0,x@0:Id.
. . . | . |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . | . |                              <ds,da,,,<k,x> : f,,,,·>
. . . | . |                           else  fi)(x@0)?Top
. . . | . 11. ∀i@0:Id, a:Action(i@0).
. . . | . |   ¬isnull(a)
. . . | . |   ⇒
. . . | . |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . | . |                                 <ds,da,,,<k,x> : f,,,,·>
. . . | . |                              else  fi))(kind(a))?Top)
. . . | . 12. ∀l:IdLnk, tg:Id.
. . . | . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . | . |                             <ds,da,,,<k,x> : f,,,,·>
. . . | . |                          else  fi))(rcv(l; tg))?Top
. . . | . 13. ∀i@0,x@0:Id.
. . . | . |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . | . |                        <ds,da,,,<k,x> : f,,,,·>
. . . | . |                     else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . . | . 14. ∀i@0:Id, t:ℕ.
. . . | . |   ¬isnull(a(i@0;t))
. . . | . |   ⇒
. . . | . |   (islocal(kind(a(i@0;t)))
```

```
. . . | . |     ⇒
. . . | . |     P != 1of(
. . . | . |         2of(2of(2of(
. . . | . |         if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | . |         else  fi))))(act(kind(a(i@0;t)))) ==> P
. . . | . |                                               (...
. . . | . |                                               ,val(...)))
. . . | . |   & (∀x@0:Id.
. . . | . |      E != 1of(
. . . | . |          2of(2of(2of(2of(
. . . | . |          if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | . |          else  fi)))))(<kind(a(i@0;t))
. . . | . |                         ,x@0>) ==> s(i@0;t+1).x@0
. . . | . |                                    =
. . . | . |                                    E
. . . | . |                                    ((λx.s(i@0;t).x)
. . . | . |                                    ,val(a(i@0;t))))
. . . | . |   & (∀l:IdLnk.
. . . | . |      if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | . |      else  fi.send(kind(a(i@0;t));l;λx.
. . . | . |      s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . | . |   & (∀x@0:Id.
. . . | . |      ¬L != 1of(
. . . | . |           2of(2of(2of(2of(2of(2of(
. . . | . |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | . |           else  fi)))))))(x@0) ==> deq-member(...;...;L)
. . . | . |      ⇒
. . . | . |      s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . | . |   & (∀l:IdLnk, tg:Id.
. . . | . |      ¬L != 1of(
. . . | . |           2of(2of(2of(2of(2of(2of(2of(
. . . | . |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | . |           else  fi))))))))(<l,tg>) ==> ...
. . . | . |      ⇒
. . . | . |      w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . | . 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . | . 16. ∀e:E.
. . . | . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . | . 17. j : Id
. . . | . 18. t : ℕ
. . . | . 19. j = i
. . . | . 20. kind(<j,t>) = k
. . . | . 21. ¬isnull(a(i;t))
. . . | . 22. valtype(<i,t>) ⊆r ma-valtype(da; kind(<i,t>))
. . . | . ⊢  valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . | . | by Subst (valtype(<i,t>) ~ valtype(i;a(i;t))) -1
. . . | . |
. . . | . |\
. . . | . | .....equality .....
. . . | . | 1. i : Id
. . . | . | 2. x : Id
. . . | . | 3. k : Knd
. . . | . | 4. ds : x:Id fp-> Type
. . . | . | 5. da : a:Knd fp-> Type
. . . | . | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
```

```
. . . | . | 7. w : World
. . . | . | 8. p : FairFifo
. . . | . | 9. FairFifo
. . . | . | 10. ∀i@0,x@0:Id.
. . . | . |     vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . | . |                              <ds,da,,,<k,x> : f,,,,·>
. . . | . |                           else  fi)(x@0)?Top
. . . | . | 11. ∀i@0:Id, a:Action(i@0).
. . . | . |     ¬isnull(a)
. . . | . |     ⇒
. . . | . |     (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . | . |                                 <ds,da,,,<k,x> : f,,,,·>
. . . | . |                              else  fi))(kind(a))?Top)
. . . | . | 12. ∀l:IdLnk, tg:Id.
. . . | . |     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . | . |                             <ds,da,,,<k,x> : f,,,,·>
. . . | . |                          else  fi))(rcv(l; tg))?Top
. . . | . | 13. ∀i@0,x@0:Id.
. . . | . |     x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . | . |                        <ds,da,,,<k,x> : f,,,,·>
. . . | . |                     else  fi)))(x@0) ==> s(i@0;0).x@0
. . . | . |                                          =
. . . | . |                                          x0
. . . | . | 14. ∀i@0:Id, t:ℕ.
. . . | . |     ¬isnull(a(i@0;t))
. . . | . |     ⇒
. . . | . |     ...
. . . | . |     & (∀x@0:Id.
. . . | . |        E != 1of(
. . . | . |            2of(2of(2of(2of(
. . . | . |            if eqof(IdDeq)(i@0,i)→
. . . | . |               <ds,da,,,<k,x> : f,,,,·>
. . . | . |            else  fi)))))(<kind(a(i@0;t))
. . . | . |                          ,x@0>) ==> s(i@0;t+1).x@0
. . . | . |                                     =
. . . | . |                                     E
. . . | . |                                     ((λx.s(i@0;t).x)
. . . | . |                                     ,val(a(i@0;t))))
. . . | . |     & (∀l:IdLnk.
. . . | . |        if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | . |        else  fi.send(kind(a(i@0;t));l;λx.
. . . | . |        s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . | . |     & (∀x@0:Id.
. . . | . |        ¬L != 1of(
. . . | . |             2of(2of(2of(2of(2of(2of(
. . . | . |             if eqof(IdDeq)(i@0,i)→
. . . | . |                <ds,da,,,<k,x> : f,,,,·>
. . . | . |             else  fi)))))))(x@0) ==> ...
. . . | . |        ⇒
. . . | . |        s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . | . |     & (∀l:IdLnk, tg:Id.
. . . | . |        ¬L != 1of(
. . . | . |             2of(2of(2of(2of(2of(2of(2of(
. . . | . |             if eqof(IdDeq)(i@0,i)→
. . . | . |                <ds,da,,,<k,x> : f,,,,·>
```

```
. . . | . |             else  fi))))))))(<l,tg>) ==> ...
. . . | . |        ⇒
. . . | . |        w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . | . | 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . | . | 16. ∀e:E.
. . . | . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . | . | 17. j : Id
. . . | . | 18. t : ℕ
. . . | . | 19. j = i
. . . | . | 20. kind(<j,t>) = k
. . . | . | 21. ¬isnull(a(i;t))
. . . | . | 22. valtype(<i,t>) ⊆r ma-valtype(da; kind(<i,t>))
. . . | . | ⊢  valtype(<i,t>) ~ valtype(i;a(i;t))
. . . | . |   by BackThru
. . . | . |      Thm* ∀i:Id, w:World, p:FairFifo, t:ℕ.
. . . | . |           ¬isnull(a(i;t))
. . . | . |           ⇒
. . . | . |           (valtype(<i,t>) ~ valtype(i;a(i;t)))
. . . | . |      {es_DASH_valtype_DASH_w_DASH_valtype}
. . . | . |
. . . | . .\
. . . | . . 1. i : Id
. . . | . . 2. x : Id
. . . | . . 3. k : Knd
. . . | . . 4. ds : x:Id fp-> Type
. . . | . . 5. da : a:Knd fp-> Type
. . . | . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . | . . 7. w : World
. . . | . . 8. p : FairFifo
. . . | . . 9. FairFifo
. . . | . . 10. ∀i@0,x@0:Id.
. . . | . . |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . | . . |                             <ds,da,,,<k,x> : f,,,,·>
. . . | . . |                          else  fi)(x@0)?Top
. . . | . . 11. ∀i@0:Id, a:Action(i@0).
. . . | . . |   ¬isnull(a)
. . . | . . |   ⇒
. . . | . . |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . | . . |                                <ds,da,,,<k,x> : f,,,,·>
. . . | . . |                             else  fi))(kind(a))?Top)
. . . | . . 12. ∀l:IdLnk, tg:Id.
. . . | . . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . | . . |                            <ds,da,,,<k,x> : f,,,,·>
. . . | . . |                         else  fi))(rcv(l; tg))?Top
. . . | . . 13. ∀i@0,x@0:Id.
. . . | . . |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . | . . |                       <ds,da,,,<k,x> : f,,,,·>
. . . | . . |                    else  fi)))(x@0) ==> s(i@0;0).x@0
. . . | . . |                                        =
. . . | . . |                                        x0
. . . | . . 14. ∀i@0:Id, t:ℕ.
. . . | . . |   ¬isnull(a(i@0;t))
. . . | . . |   ⇒
. . . | . . |   ...
. . . | . . |   & (∀x@0:Id.
```

```
. . . | . . |      E != 1of(
. . . | . . |           2of(2of(2of(2of(
. . . | . . |           if eqof(IdDeq)(i@0,i)→
. . . | . . |              <ds,da,,,<k,x> : f,,,,·>
. . . | . . |           else  fi)))))(<kind(a(i@0;t))
. . . | . . |                         ,x@0>) ==> s(i@0;t+1).x@0
. . . | . . |                                    =
. . . | . . |                                    E
. . . | . . |                                    ((λx.s(i@0;t).x)
. . . | . . |                                    ,val(a(i@0;t))))
. . . | . . |   & (∀l:IdLnk.
. . . | . . |      if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | . . |      else  fi.send(kind(a(i@0;t));l;λx.
. . . | . . |      s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . | . . |   & (∀x@0:Id.
. . . | . . |      ¬L != 1of(
. . . | . . |            2of(2of(2of(2of(2of(2of(
. . . | . . |            if eqof(IdDeq)(i@0,i)→
. . . | . . |               <ds,da,,,<k,x> : f,,,,·>
. . . | . . |            else  fi)))))))(x@0) ==> ...
. . . | . . |      ⇒
. . . | . . |      s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . | . . |   & (∀l:IdLnk, tg:Id.
. . . | . . |      ¬L != 1of(
. . . | . . |            2of(2of(2of(2of(2of(2of(2of(
. . . | . . |            if eqof(IdDeq)(i@0,i)→
. . . | . . |               <ds,da,,,<k,x> : f,,,,·>
. . . | . . |            else  fi))))))))(<l,tg>) ==> ...
. . . | . . |      ⇒
. . . | . . |      w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . | . . 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . | . . 16. ∀e:E.
. . . | . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . | . . 17. j : Id
. . . | . . 18. t : ℕ
. . . | . . 19. j = i
. . . | . . 20. kind(<j,t>) = k
. . . | . . 21. ¬isnull(a(i;t))
. . . | . . 22. valtype(i;a(i;t)) ⊆r ma-valtype(da; kind(<i,t>))
. . . | . . ⊢  valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . | . . | by Subst (kind(<i,t>) = k) -1
. . . | . . |
. . . | . . .....equality .....
. . . | . . 1. i : Id
. . . | . . 2. x : Id
. . . | . . 3. k : Knd
. . . | . . 4. ds : x:Id fp-> Type
. . . | . . 5. da : a:Knd fp-> Type
. . . | . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . | . . 7. w : World
. . . | . . 8. p : FairFifo
. . . | . . 9. FairFifo
. . . | . . 10. ∀i@0,x@0:Id.
. . . | . . |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . | . . |                              <ds,da,,,<k,x> : f,,,,·>
```

```
. . . | . . |                           else  fi)(x@0)?Top
. . . | . . 11. ∀i@0:Id, a:Action(i@0).
. . . | . . |   ¬isnull(a)
. . . | . . |   ⇒
. . . | . . |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . | . . |                                 <ds,da,,,<k,x> : f,,,,·>
. . . | . . |                              else  fi))(kind(a))?Top)
. . . | . . 12. ∀l:IdLnk, tg:Id.
. . . | . . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . | . . |                             <ds,da,,,<k,x> : f,,,,·>
. . . | . . |                         else  fi))(rcv(l; tg))?Top
. . . | . . 13. ∀i@0,x@0:Id.
. . . | . . |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . | . . |                        <ds,da,,,<k,x> : f,,,,·>
. . . | . . |                    else  fi)))(x@0) ==> s(i@0;0).x@0
. . . | . . |                                         =
. . . | . . |                                         x0
. . . | . . 14. ∀i@0:Id, t:ℕ.
. . . | . . |   ¬isnull(a(i@0;t))
. . . | . . |   ⇒
. . . | . . |   ...
. . . | . . |   & (∀x@0:Id.
. . . | . . |      E != 1of(
. . . | . . |           2of(2of(2of(2of(
. . . | . . |           if eqof(IdDeq)(i@0,i)→
. . . | . . |              <ds,da,,,<k,x> : f,,,,·>
. . . | . . |           else  fi)))))(<kind(a(i@0;t))
. . . | . . |                         ,x@0>) ==> s(i@0;t+1).x@0
. . . | . . |                                   =
. . . | . . |                                   E
. . . | . . |                                   ((λx.s(i@0;t).x)
. . . | . . |                                   ,val(a(i@0;t))))
. . . | . . |   & (∀l:IdLnk.
. . . | . . |      if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | . . |      else  fi.send(kind(a(i@0;t));l;λx.
. . . | . . |      s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . | . . |   & (∀x@0:Id.
. . . | . . |      ¬L != 1of(
. . . | . . |            2of(2of(2of(2of(2of(2of(
. . . | . . |            if eqof(IdDeq)(i@0,i)→
. . . | . . |               <ds,da,,,<k,x> : f,,,,·>
. . . | . . |            else  fi)))))))(x@0) ==> ...

. . . | . . |      ⇒
. . . | . . |      s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . | . . |   & (∀l:IdLnk, tg:Id.
. . . | . . |      ¬L != 1of(
. . . | . . |            2of(2of(2of(2of(2of(2of(2of(
. . . | . . |            if eqof(IdDeq)(i@0,i)→
. . . | . . |               <ds,da,,,<k,x> : f,,,,·>
. . . | . . |            else  fi))))))))(<l,tg>) ==> ...
. . . | . . |      ⇒
. . . | . . |      w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . | . . 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . | . . 16. ∀e:E.
. . . | . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
```

```
. . . | . . 17. j : Id
. . . | . . 18. t : ℕ
. . . | . . 19. j = i
. . . | . . 20. kind(<j,t>) = k
. . . | . . 21. ¬isnull(a(i;t))
. . . | . . 22. valtype(i;a(i;t)) ⊆r ma-valtype(da; kind(<i,t>))
. . . | . . ⊢  kind(<i,t>) = k
. . . | . . | by ES_Reduce THEN NthHypEq -3 THEN RepeatFor 2 Analyze
. . . | . . |
. . . | . . .....subterm .....
. . . | . . 1. i : Id
. . . | . . 2. x : Id
. . . | . . 3. k : Knd
. . . | . . 4. ds : x:Id fp-> Type
. . . | . . 5. da : a:Knd fp-> Type
. . . | . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . | . . 7. w : World
. . . | . . 8. p : FairFifo
. . . | . . 9. FairFifo
. . . | . . 10. ∀i@0,x@0:Id.
. . . | . .     vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . | . .                                <ds,da,,,<k,x> : f,,,,·>
. . . | . .                             else  fi)(x@0)?Top
. . . | . . 11. ∀i@0:Id, a:Action(i@0).
. . . | . .     ¬isnull(a)
. . . | . .     ⇒
. . . | . .     (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . | . .                                   <ds,da,,,<k,x> : f,,,,·>
. . . | . .                                else  fi))(kind(a))?Top)
. . . | . . 12. ∀l:IdLnk, tg:Id.
. . . | . .     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . | . .                               <ds,da,,,<k,x> : f,,,,·>
. . . | . .                            else  fi))(rcv(l; tg))?Top
. . . | . . 13. ∀i@0,x@0:Id.
. . . | . .     x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . | . .                          <ds,da,,,<k,x> : f,,,,·>
. . . | . .                       else  fi)))(x@0) ==> s(i@0;0).x@0
. . . | . .                                            =
. . . | . .                                            x0
. . . | . . 14. ∀i@0:Id, t:ℕ.
. . . | . .     ¬isnull(a(i@0;t))
. . . | . .     ⇒
. . . | . .     ...
. . . | . .     & (∀x@0:Id.
. . . | . .        E != 1of(
. . . | . .             2of(2of(2of(2of(
. . . | . .             if eqof(IdDeq)(i@0,i)→
. . . | . .                <ds,da,,,<k,x> : f,,,,·>
. . . | . .             else  fi)))))(<kind(a(i@0;t))
. . . | . .                           ,x@0>) ==> s(i@0;t+1).x@0
. . . | . .                                      =
. . . | . .                                      E
. . . | . .                                      ((λx.s(i@0;t).x)
. . . | . .                                      ,val(a(i@0;t))))
. . . | . .     & (∀l:IdLnk.
```

```
. . . | . .         if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . | . .         else  fi.send(kind(a(i@0;t));l;λx.
. . . | . .         s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . | . .      & (∀x@0:Id.
. . . | . .         ¬L != 1of(
. . . | . .               2of(2of(2of(2of(2of(2of(
. . . | . .               if eqof(IdDeq)(i@0,i)→
. . . | . .                  <ds,da,,,<k,x> : f,,,,·>
. . . | . .               else  fi)))))))(x@0) ==> ...
. . . | . .         ⇒
. . . | . .         s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . | . .      & (∀l:IdLnk, tg:Id.
. . . | . .         ¬L != 1of(
. . . | . .               2of(2of(2of(2of(2of(2of(2of(
. . . | . .               if eqof(IdDeq)(i@0,i)→
. . . | . .                  <ds,da,,,<k,x> : f,,,,·>
. . . | . .               else  fi))))))))(<l,tg>) ==> ...
. . . | . .         ⇒
. . . | . .         w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . | . . 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . | . . 16. ∀e:E.
. . . | . .     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . | . . 17. j : Id
. . . | . . 18. t : ℕ
. . . | . . 19. j = i
. . . | . . 20. kind(<j,t>) = k
. . . | . . 21. ¬isnull(a(i;t))
. . . | . . 22. valtype(i;a(i;t)) ⊆r ma-valtype(da; kind(<i,t>))
. . . | . . ⊢  <i,t> = <j,t>
. . . | . .   by Unfold ‘w-E‘ 0 THEN Analyze THEN Reduce 0
. . . | . .      THEN
. . . | . .      Analyze

. . . |
. . . .\
. . . . 1. i : Id
. . . . 2. x : Id
. . . . 3. k : Knd
. . . . 4. ds : x:Id fp-> Type
. . . . 5. da : a:Knd fp-> Type
. . . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . . 7. w : World
. . . . 8. p : FairFifo
. . . . 9. FairFifo
. . . . 10. ∀i@0,x@0:Id.
. . . . |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . . |                                <ds,da,,,<k,x> : f,,,,·>
. . . . |                             else  fi)(x@0)?Top
. . . . 11. ∀i@0:Id, a:Action(i@0).
. . . . |   ¬isnull(a)
. . . . |   ⇒
. . . . |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . . |                                   <ds,da,,,<k,x> : f,,,,·>
. . . . |                                else  fi))(kind(a))?Top)
. . . . 12. ∀l:IdLnk, tg:Id.
. . . . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
```

```
. . . . |                             <ds,da,,,<k,x> : f,,,,·>
. . . . |                          else  fi))(rcv(l; tg))?Top
. . . . 13. ∀i@0,x@0:Id.
. . . . |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . . |                          <ds,da,,,<k,x> : f,,,,·>
. . . . |                       else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . . . 14. ∀i@0:Id, t:ℕ.
. . . . |   ¬isnull(a(i@0;t))
. . . . |   ⇒
. . . . |   (islocal(kind(a(i@0;t)))
. . . . |    ⇒
. . . . |    P != 1of(
. . . . |         2of(2of(2of(
. . . . |         if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |         else  fi))))(act(kind(a(i@0;t)))) ==> ...)
. . . . |   & (∀x@0:Id.
. . . . |      E != 1of(
. . . . |           2of(2of(2of(2of(
. . . . |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |           else  fi)))))(<kind(a(i@0;t))
. . . . |                         ,x@0>) ==> s(i@0;t+1).x@0
. . . . |                                    =
. . . . |                                    E
. . . . |                                    ((λx.s(i@0;t).x)
. . . . |                                    ,val(a(i@0;t))))
. . . . |   & (∀l:IdLnk.
. . . . |      if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |      else  fi.send(kind(a(i@0;t));l;λx.
. . . . |      s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . . |   & (∀x@0:Id.
. . . . |      ¬L != 1of(
. . . . |            2of(2of(2of(2of(2of(2of(
. . . . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |            else  fi)))))))(x@0) ==> deq-member(...;...;L)
. . . . |      ⇒
. . . . |      s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . . |   & (∀l:IdLnk, tg:Id.
. . . . |      ¬L != 1of(
. . . . |            2of(2of(2of(2of(2of(2of(2of(
. . . . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |            else  fi))))))))(<l,tg>) ==> ...
. . . . |      ⇒
. . . . |      w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . . 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . . 16. ∀e:E.
. . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . . 17. j : Id
. . . . 18. t : ℕ
. . . . 19. j = i
. . . . 20. kind(<j,t>) = k
. . . . 21. ¬isnull(a(i;t))
. . . . 22. valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . . ⊢  (x after <j,t>) = f((λz.(z when <j,t>)),val(<j,t>))
. . . . | by MoveToHyp -4 -1
. . . . |    THEN
```

```
. . . . |    AllHyps (λh.InstHyp [i;t] h THENA Complete Auto)
. . . . |    THEN
. . . . |    Subst ((eqof(IdDeq)(i,i)) ~ true₂) -1
. . . . |    THEN
. . . . |    Reduce -1
. . . . |    THEN
. . . . |    Try
. . . . |    (BackThru
. . . . |     Thm* ∀d:EqDecider(A), i:A. (eqof(d)(i,i)) ~ true₂
. . . . |     {eqof_eq_btrue})
. . . . |    THEN
. . . . |    ExRepD
. . . . |
. . . . 1. i : Id
. . . . 2. x : Id
. . . . 3. k : Knd
. . . . 4. ds : x:Id fp-> Type
. . . . 5. da : a:Knd fp-> Type
. . . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . . 7. w : World
. . . . 8. p : FairFifo
. . . . 9. FairFifo
. . . . 10. ∀i@0,x@0:Id.
. . . . |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . . |                              <ds,da,,,<k,x> : f,,,,·>
. . . . |                           else  fi)(x@0)?Top
. . . . 11. ∀i@0:Id, a:Action(i@0).
. . . . |   ¬isnull(a)
. . . . |   ⇒
. . . . |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . . |                                 <ds,da,,,<k,x> : f,,,,·>
. . . . |                              else  fi))(kind(a))?Top)
. . . . 12. ∀l:IdLnk, tg:Id.
. . . . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . . |                             <ds,da,,,<k,x> : f,,,,·>
. . . . |                          else  fi))(rcv(l; tg))?Top
. . . . 13. ∀i@0,x@0:Id.
. . . . |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . . |                        <ds,da,,,<k,x> : f,,,,·>
. . . . |                     else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . . . 14. ∀i@0:Id, t:ℕ.
. . . . |   ¬isnull(a(i@0;t))
. . . . |   ⇒
. . . . |   (islocal(kind(a(i@0;t)))
. . . . |    ⇒
. . . . |    P != 1of(
. . . . |         2of(2of(2of(
. . . . |         if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |         else  fi))))(act(kind(a(i@0;t)))) ==> ...)
. . . . |   & (∀x@0:Id.
. . . . |      E != 1of(
. . . . |           2of(2of(2of(2of(
. . . . |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |           else  fi)))))(<kind(a(i@0;t))
. . . . |                         ,x@0>) ==> s(i@0;t+1).x@0
```

```
. . . . |                                        =
. . . . |                                        E
. . . . |                                        ((λx.s(i@0;t).x)
. . . . |                                        ,val(a(i@0;t))))
. . . . |   & (∀l:IdLnk.
. . . . |      if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |      else  fi.send(kind(a(i@0;t));l;λx.
. . . . |      s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . . |   & (∀x@0:Id.
. . . . |      ¬L != 1of(
. . . . |            2of(2of(2of(2of(2of(2of(
. . . . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |            else  fi)))))))(x@0) ==> deq-member(...;...;L)
. . . . |      ⇒
. . . . |      s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . . |   & (∀l:IdLnk, tg:Id.
. . . . |      ¬L != 1of(
. . . . |            2of(2of(2of(2of(2of(2of(2of(
. . . . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |            else  fi))))))))(<l,tg>) ==> ...
. . . . |      ⇒
. . . . |      w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . . 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . . 16. ∀e:E.
. . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . . 17. j : Id
. . . . 18. t : ℕ
. . . . 19. valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . . 20. j = i
. . . . 21. kind(<j,t>) = k
. . . . 22. ¬isnull(a(i;t))
. . . . 23. islocal(kind(a(i;t)))
. . . . |   ⇒
. . . . |   P != (act(kind(a(i;t)))) ==> P((λx.s(i;t).x),val(a(i;t)))
. . . . 24. ∀x@0:Id.
. . . . |   E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. . . . |                                              =
. . . . |                                              E
. . . . |                                              ((λx.s(i;t).x)
. . . . |                                              ,val(a(i;t)))
. . . . 25. ∀l:IdLnk.
. . . . |   <ds,da,,,<k,x> : f,,,,·>.send(kind(a(i;t));l;λx.
. . . . |   s(i;t).x;val(a(i;t));withlnk(l;m(i;t));i)
. . . . 26. ∀x@0:Id.
. . . . |   ¬L != (x@0) ==> deq-member(KindDeq;kind(a(i;t));L)
. . . . |   ⇒
. . . . |   s(i;t).x@0 = s(i;t+1).x@0
. . . . 27. ∀l:IdLnk, tg:Id.
. . . . |   ¬L != (<l,tg>) ==> deq-member(KindDeq;kind(a(i;t));L)
. . . . |   ⇒
. . . . |   w-tagged(tg; onlnk(l;m(i;t))) = nil
. . . . ⊢  (x after <j,t>) = f((λz.(z when <j,t>)),val(<j,t>))
. . . . | by RepeatFor 3 (Thin -1) THEN InstHyp [x] -1
. . . . |    THEN
. . . . |    Unfold `fpf-val` -1
```

```
. . . . |    THEN
. . . . |    Analyze -1
. . . . |
. . . . |\
. . . . | .....antecedent .....
. . . . | 1. i : Id
. . . . | 2. x : Id
. . . . | 3. k : Knd
. . . . | 4. ds : x:Id fp-> Type
. . . . | 5. da : a:Knd fp-> Type
. . . . | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . . | 7. w : World
. . . . | 8. p : FairFifo
. . . . | 9. FairFifo
. . . . | 10. ∀i@0,x@0:Id.
. . . . | |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . . | |                            <ds,da,,,<k,x> : f,,,,·>
. . . . | |                         else  fi)(x@0)?Top
. . . . | 11. ∀i@0:Id, a:Action(i@0).
. . . . | |   ¬isnull(a)
. . . . | |   ⇒
. . . . | |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . . | |                               <ds,da,,,<k,x> : f,,,,·>
. . . . | |                            else  fi))(kind(a))?Top)
. . . . | 12. ∀l:IdLnk, tg:Id.
. . . . | |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . . | |                           <ds,da,,,<k,x> : f,,,,·>
. . . . | |                        else  fi))(rcv(l; tg))?Top
. . . . | 13. ∀i@0,x@0:Id.
. . . . | |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . . | |                      <ds,da,,,<k,x> : f,,,,·>
. . . . | |                   else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . . . | 14. ∀i@0:Id, t:ℕ.
. . . . | |   ¬isnull(a(i@0;t))
. . . . | |   ⇒
. . . . | |   (islocal(kind(a(i@0;t)))
. . . . | |    ⇒
. . . . | |    P != 1of(
. . . . | |         2of(2of(2of(
. . . . | |         if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . | |         else  fi))))(act(kind(a(i@0;t)))) ==> P
. . . . | |                                              (...
. . . . | |                                              ,val(...)))
. . . . | |   & (∀x@0:Id.
. . . . | |      E != 1of(
. . . . | |           2of(2of(2of(2of(
. . . . | |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . | |           else  fi)))))(<kind(a(i@0;t))
. . . . | |                         ,x@0>) ==> s(i@0;t+1).x@0
. . . . | |                                    =
. . . . | |                                    E
. . . . | |                                    ((λx.s(i@0;t).x)
. . . . | |                                    ,val(a(i@0;t))))
. . . . | |   & (∀l:IdLnk.
. . . . | |      if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
```

```
. . . . | |      else  fi.send(kind(a(i@0;t));l;λx.
. . . . | |      s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . . | |   & (∀x@0:Id.
. . . . | |      ¬L != 1of(
. . . . | |            2of(2of(2of(2of(2of(2of(
. . . . | |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . | |            else  fi)))))))(x@0) ==> deq-member(...;...;L)
. . . . | |      ⇒
. . . . | |      s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . . | |   & (∀l:IdLnk, tg:Id.
. . . . | |      ¬L != 1of(
. . . . | |            2of(2of(2of(2of(2of(2of(2of(
. . . . | |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . | |            else  fi))))))))(<l,tg>) ==> ...
. . . . | |      ⇒
. . . . | |      w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . . | 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . . | 16. ∀e:E.
. . . . | |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . . | 17. j : Id
. . . . | 18. t : ℕ
. . . . | 19. valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . . | 20. j = i
. . . . | 21. kind(<j,t>) = k
. . . . | 22. ¬isnull(a(i;t))
. . . . | 23. islocal(kind(a(i;t)))
. . . . | |   ⇒
. . . . | |   P != (act(kind(a(i;t)))) ==> P
. . . . | |                                 ((λx.s(i;t).x)
. . . . | |                                 ,val(a(i;t)))
. . . . | 24. ∀x@0:Id.
. . . . | |   E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. . . . | |                                          =
. . . . | |                                          E
. . . . | |                                          ((λx.s(i;t).x)
. . . . | |                                          ,val(a(i;t)))
. . . . | ⊢  <kind(a(i;t)),x> ∈ dom(<k,x> : f)
. . . . | | by ((Unfolds [‘fpf-dom‘;‘fpf-single‘] 0 THEN Reduce 0
. . . . | |      THEN
. . . . | |      RW assert_pushdownC 0)
. . . . | |     THEN
. . . . | |     (RWO "deq_property<" 0)
. . . . | |     THEN
. . . . | |     OrLeft)
. . . . | |    THEN
. . . . | |    Analyze
. . . . | |
. . . . | .....subterm .....
. . . . | 1. i : Id
. . . . | 2. x : Id
. . . . | 3. k : Knd
. . . . | 4. ds : x:Id fp-> Type
. . . . | 5. da : a:Knd fp-> Type
. . . . | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . . | 7. w : World
```

```
. . . . | 8. p : FairFifo
. . . . | 9. FairFifo
. . . . | 10. ∀i@0,x@0:Id.
. . . . |     vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . . |                               <ds,da,,,<k,x> : f,,,,·>
. . . . |                            else  fi)(x@0)?Top
. . . . | 11. ∀i@0:Id, a:Action(i@0).
. . . . |     ¬isnull(a)
. . . . |     ⇒
. . . . |     (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . . |                                  <ds,da,,,<k,x> : f,,,,·>
. . . . |                               else  fi))(kind(a))?Top)
. . . . | 12. ∀l:IdLnk, tg:Id.
. . . . |     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . . |                              <ds,da,,,<k,x> : f,,,,·>
. . . . |                           else  fi))(rcv(l; tg))?Top
. . . . | 13. ∀i@0,x@0:Id.
. . . . |     x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . . |                         <ds,da,,,<k,x> : f,,,,·>
. . . . |                      else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . . . | 14. ∀i@0:Id, t:ℕ.
. . . . |     ¬isnull(a(i@0;t))
. . . . |     ⇒
. . . . |     (islocal(kind(a(i@0;t)))
. . . . |      ⇒
. . . . |      P != 1of(
. . . . |           2of(2of(2of(
. . . . |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |           else  fi))))(act(kind(a(i@0;t)))) ==> P
. . . . |                                                (...
. . . . |                                                ,val(...)))
. . . . |     & (∀x@0:Id.
. . . . |        E != 1of(
. . . . |             2of(2of(2of(2of(
. . . . |             if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |             else  fi)))))(<kind(a(i@0;t))
. . . . |                         ,x@0>) ==> s(i@0;t+1).x@0
. . . . |                                    =
. . . . |                                    E
. . . . |                                    ((λx.s(i@0;t).x)
. . . . |                                    ,val(a(i@0;t))))
. . . . |     & (∀l:IdLnk.
. . . . |        if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |        else  fi.send(kind(a(i@0;t));l;λx.
. . . . |        s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . . |     & (∀x@0:Id.
. . . . |        ¬L != 1of(
. . . . |              2of(2of(2of(2of(2of(2of(
. . . . |              if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |              else  fi)))))))(x@0) ==> deq-member(...;...;L)
. . . . |        ⇒
. . . . |        s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . . |     & (∀l:IdLnk, tg:Id.
. . . . |        ¬L != 1of(
. . . . |              2of(2of(2of(2of(2of(2of(2of(
```

```
. . . . |              if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . |              else  fi))))))))(<l,tg>) ==> ...
. . . . |        ⇒
. . . . |        w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . . | 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . . | 16. ∀e:E.
. . . . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . . | 17. j : Id
. . . . | 18. t : ℕ
. . . . | 19. valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . . | 20. j = i
. . . . | 21. kind(<j,t>) = k
. . . . | 22. ¬isnull(a(i;t))
. . . . | 23. islocal(kind(a(i;t)))
. . . . |     ⇒
. . . . |     P != (act(kind(a(i;t)))) ==> P
. . . . |                                  ((λx.s(i;t).x)
. . . . |                                  ,val(a(i;t)))
. . . . | 24. ∀x@0:Id.
. . . . |     E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. . . . |                                            =
. . . . |                                            E
. . . . |                                            ((λx.s(i;t).x)
. . . . |                                            ,val(a(i;t)))
. . . . | ⊢  k = kind(a(i;t))
. . . . |   by Repeat (Unfolds [‘w-ekind‘;‘w-act‘] -4 THEN Reduce -4)
. . . . |      THEN
. . . . |      Assert (kind(a(i;t)) = kind(a(j;t)))
. . . . |      THEN
. . . . |      Analyze
. . . . |      THEN
. . . . |      Analyze
. . . . |
. . . . .\
. . . . . 1. i : Id
. . . . . 2. x : Id
. . . . . 3. k : Knd
. . . . . 4. ds : x:Id fp-> Type
. . . . . 5. da : a:Knd fp-> Type
. . . . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . . . 7. w : World
. . . . . 8. p : FairFifo
. . . . . 9. FairFifo
. . . . . 10. ∀i@0,x@0:Id.
. . . . . |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . . . |                              <ds,da,,,<k,x> : f,,,,·>
. . . . . |                           else  fi)(x@0)?Top
. . . . . 11. ∀i@0:Id, a:Action(i@0).
. . . . . |   ¬isnull(a)
. . . . . |   ⇒
. . . . . |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . . . |                                 <ds,da,,,<k,x> : f,,,,·>
. . . . . |                              else  fi))(kind(a))?Top)
. . . . . 12. ∀l:IdLnk, tg:Id.
. . . . . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
```

```
. . . . . |                             <ds,da,,,<k,x> : f,,,,·>
. . . . . |                          else  fi))(rcv(l; tg))?Top
. . . . . 13. ∀i@0,x@0:Id.
. . . . . |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . . . |                        <ds,da,,,<k,x> : f,,,,·>
. . . . . |                     else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . . . . 14. ∀i@0:Id, t:ℕ.
. . . . . |   ¬isnull(a(i@0;t))
. . . . . |   ⇒
. . . . . |   (islocal(kind(a(i@0;t)))
. . . . . |    ⇒
. . . . . |    P != 1of(
. . . . . |         2of(2of(2of(
. . . . . |         if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . . |         else  fi))))(act(kind(a(i@0;t)))) ==> P
. . . . . |                                               (...
. . . . . |                                               ,val(...)))
. . . . . |   & (∀x@0:Id.
. . . . . |      E != 1of(
. . . . . |           2of(2of(2of(2of(
. . . . . |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . . |           else  fi)))))(<kind(a(i@0;t))
. . . . . |                         ,x@0>) ==> s(i@0;t+1).x@0
. . . . . |                                    =
. . . . . |                                    E
. . . . . |                                    ((λx.s(i@0;t).x)
. . . . . |                                    ,val(a(i@0;t))))
. . . . . |   & (∀l:IdLnk.
. . . . . |      if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . . |      else  fi.send(kind(a(i@0;t));l;λx.
. . . . . |      s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . . . |   & (∀x@0:Id.
. . . . . |      ¬L != 1of(
. . . . . |            2of(2of(2of(2of(2of(2of(
. . . . . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . . |            else  fi)))))))(x@0) ==> deq-member(...;...;L)
. . . . . |      ⇒
. . . . . |      s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . . . |   & (∀l:IdLnk, tg:Id.
. . . . . |      ¬L != 1of(
. . . . . |            2of(2of(2of(2of(2of(2of(2of(
. . . . . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . . |            else  fi))))))))(<l,tg>) ==> ...
. . . . . |      ⇒
. . . . . |      w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . . . 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . . . 16. ∀e:E.
. . . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . . . 17. j : Id
. . . . . 18. t : ℕ
. . . . . 19. valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . . . 20. j = i
. . . . . 21. kind(<j,t>) = k
. . . . . 22. ¬isnull(a(i;t))
. . . . . 23. islocal(kind(a(i;t)))
```

```
. . . . . |   ⇒
. . . . . |   P != (act(kind(a(i;t)))) ==> P
. . . . . |                                 ((λx.s(i;t).x)
. . . . . |                                 ,val(a(i;t)))
. . . . . 24. ∀x@0:Id.
. . . . . |   E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. . . . . |                                          =
. . . . . |                                          E
. . . . . |                                          ((λx.s(i;t).x)
. . . . . |                                          ,val(a(i;t)))
. . . . . 25. s(i;t+1).x
. . . . . |   =
. . . . . |   <k,x> : f(<kind(a(i;t)),x>)((λx.s(i;t).x),val(a(i;t)))
. . . . . ⊢  (x after <j,t>) = f((λz.(z when <j,t>)),val(<j,t>))
. . . . . | by Unfolds [‘fpf-single‘;‘fpf-ap‘] -1 THEN Reduce -1
. . . . . |    THEN
. . . . . |    Repeat
. . . . . |    (Unfolds [‘w-after‘;‘w-when‘;‘es-val‘;‘w-eval‘;‘w-act‘]
. . . . . |     0
. . . . . |     THEN
. . . . . |     Reduce 0)
. . . . . |
. . . . . 1. i : Id
. . . . . 2. x : Id
. . . . . 3. k : Knd
. . . . . 4. ds : x:Id fp-> Type
. . . . . 5. da : a:Knd fp-> Type
. . . . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . . . 7. w : World
. . . . . 8. p : FairFifo
. . . . . 9. FairFifo
. . . . . 10. ∀i@0,x@0:Id.
. . . . . |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . . . |                              <ds,da,,,<k,x> : f,,,,·>
. . . . . |                           else  fi)(x@0)?Top
. . . . . 11. ∀i@0:Id, a:Action(i@0).
. . . . . |   ¬isnull(a)
. . . . . |   ⇒
. . . . . |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . . . |                                 <ds,da,,,<k,x> : f,,,,·>
. . . . . |                              else  fi))(kind(a))?Top)
. . . . . 12. ∀l:IdLnk, tg:Id.
. . . . . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . . . |                             <ds,da,,,<k,x> : f,,,,·>
. . . . . |                          else  fi))(rcv(l; tg))?Top
. . . . . 13. ∀i@0,x@0:Id.
. . . . . |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . . . |                        <ds,da,,,<k,x> : f,,,,·>
. . . . . |                     else  fi)))(x@0) ==> s(i@0;0).x@0 = x0
. . . . . 14. ∀i@0:Id, t:ℕ.
. . . . . |   ¬isnull(a(i@0;t))
. . . . . |   ⇒
. . . . . |   (islocal(kind(a(i@0;t)))
. . . . . |    ⇒
. . . . . |    P != 1of(
```

```
. . . . . |         2of(2of(2of(
. . . . . |         if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . . |         else  fi))))(act(kind(a(i@0;t)))) ==> P
. . . . . |                                               (...
. . . . . |                                               ,val(...)))
. . . . . |   & (∀x@0:Id.
. . . . . |      E != 1of(
. . . . . |           2of(2of(2of(2of(
. . . . . |           if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . . |           else  fi)))))(<kind(a(i@0;t))
. . . . . |                         ,x@0>) ==> s(i@0;t+1).x@0
. . . . . |                                    =
. . . . . |                                    E
. . . . . |                                    ((λx.s(i@0;t).x)
. . . . . |                                    ,val(a(i@0;t))))
. . . . . |   & (∀l:IdLnk.
. . . . . |      if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . . |      else  fi.send(kind(a(i@0;t));l;λx.
. . . . . |      s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . . . |   & (∀x@0:Id.
. . . . . |      ¬L != 1of(
. . . . . |            2of(2of(2of(2of(2of(2of(
. . . . . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . . |            else  fi)))))))(x@0) ==> deq-member(...;...;L)
. . . . . |      ⇒
. . . . . |      s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . . . |   & (∀l:IdLnk, tg:Id.
. . . . . |      ¬L != 1of(
. . . . . |            2of(2of(2of(2of(2of(2of(2of(
. . . . . |            if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . . |            else  fi))))))))(<l,tg>) ==> ...
. . . . . |      ⇒
. . . . . |      w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . . . 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . . . 16. ∀e:E.
. . . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . . . 17. j : Id
. . . . . 18. t : ℕ
. . . . . 19. valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . . . 20. j = i
. . . . . 21. kind(<j,t>) = k
. . . . . 22. ¬isnull(a(i;t))
. . . . . 23. islocal(kind(a(i;t)))
. . . . . |   ⇒
. . . . . |   P != (act(kind(a(i;t)))) ==> P
. . . . . |                                ((λx.s(i;t).x)
. . . . . |                                ,val(a(i;t)))
. . . . . 24. ∀x@0:Id.
. . . . . |   E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. . . . . |                                          =
. . . . . |                                          E
. . . . . |                                          ((λx.s(i;t).x)
. . . . . |                                          ,val(a(i;t)))
. . . . . 25. s(i;t+1).x = f((λx.s(i;t).x),val(a(i;t)))
. . . . . ⊢  s(j;t+1).x = f((λz.s(j;t).z),val(a(j;t)))
```

```
. . . . . | by AllHyps ES_ReduceHyp THEN StrongHypSubst -6 0
. . . . . |
. . . . . |\
. . . . . | .....subterm .....
. . . . . | 1. i : Id
. . . . . | 2. x : Id
. . . . . | 3. k : Knd
. . . . . | 4. ds : x:Id fp-> Type
. . . . . | 5. da : a:Knd fp-> Type
. . . . . | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . . . | 7. w : World
. . . . . | 8. p : FairFifo
. . . . . | 9. FairFifo
. . . . . | 10. ∀i@0,x@0:Id.
. . . . . |     vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . . . |                               <ds,da,,,<k,x> : f,,,,·>
. . . . . |                            else  fi)(x@0)?Top
. . . . . | 11. ∀i@0:Id, a:Action(i@0).
. . . . . |     ¬isnull(a)
. . . . . |     ⇒
. . . . . |     (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . . . |                                   <ds,da,,,<k,x> : f,,,,·>
. . . . . |                                else  fi))(kind(a))?Top)
. . . . . | 12. ∀l:IdLnk, tg:Id.
. . . . . |     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . . . |                              <ds,da,,,<k,x> : f,,,,·>
. . . . . |                           else  fi))(rcv(l; tg))?Top
. . . . . | 13. ∀i@0,x@0:Id.
. . . . . |     x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . . . |                         <ds,da,,,<k,x> : f,,,,·>
. . . . . |                      else  fi)))(x@0) ==> s(i@0;0).x@0
. . . . . |                                           =
. . . . . |                                           x0
. . . . . | 14. ∀i@0:Id, t:ℕ.
. . . . . |     ¬isnull(a(i@0;t))
. . . . . |     ⇒
. . . . . |     ...
. . . . . |     & (∀x@0:Id.
. . . . . |        E != 1of(
. . . . . |             2of(2of(2of(2of(
. . . . . |             if eqof(IdDeq)(i@0,i)→
. . . . . |                <ds,da,,,<k,x> : f,,,,·>
. . . . . |             else  fi)))))(<kind(a(i@0;t))
. . . . . |                           ,x@0>) ==> s(i@0;t+1).x@0
. . . . . |                                      =
. . . . . |                                      E
. . . . . |                                      ((λx.s(i@0;t).x)
. . . . . |                                      ,val(a(i@0;t))))
. . . . . |     & (∀l:IdLnk.
. . . . . |        if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . . |        else  fi.send(kind(a(i@0;t));l;λx.
. . . . . |        s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . . . |     & (∀x@0:Id.
. . . . . |        ¬L != 1of(
. . . . . |              2of(2of(2of(2of(2of(2of(
```

```
. . . . . |              if eqof(IdDeq)(i@0,i)→
. . . . . |                 <ds,da,,,<k,x> : f,,,,·>
. . . . . |              else  fi)))))))(x@0) ==> ...
. . . . . |        ⇒
. . . . . |        s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . . . |     & (∀l:IdLnk, tg:Id.
. . . . . |        ¬L != 1of(
. . . . . |              2of(2of(2of(2of(2of(2of(2of(
. . . . . |              if eqof(IdDeq)(i@0,i)→
. . . . . |                 <ds,da,,,<k,x> : f,,,,·>
. . . . . |              else  fi))))))))(<l,tg>) ==> ...
. . . . . |        ⇒
. . . . . |        w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . . . | 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . . . | 16. ∀e:E.
. . . . . |     loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . . . | 17. j : Id
. . . . . | 18. t : ℕ
. . . . . | 19. valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . . . | 20. j = i
. . . . . | 21. kind(<j,t>) = k
. . . . . | 22. ¬isnull(a(i;t))
. . . . . | 23. islocal(kind(a(i;t)))
. . . . . |     ⇒
. . . . . |     P != (act(kind(a(i;t)))) ==> P
. . . . . |                                  ((λx.s(i;t).x)
. . . . . |                                  ,val(a(i;t)))
. . . . . | 24. ∀x@0:Id.
. . . . . |     E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. . . . . |                                            =
. . . . . |                                            E
. . . . . |                                            ((λx.s(i;t).x)
. . . . . |                                            ,val(a(i;t)))
. . . . . | 25. s(i;t+1).x = f((λx.s(i;t).x),val(a(i;t)))
. . . . . | 26. z : Id
. . . . . | 27. z = i
. . . . . | ⊢  s(z;t+1).x ∈ ds(x)?Top
. . . . . |   by DoSubsume THEN HypSubst -1 0 THEN BackThruSomeHyp
. . . . . |
. . . . . .\
. . . . . . .....subterm .....
. . . . . . 1. i : Id
. . . . . . 2. x : Id
. . . . . . 3. k : Knd
. . . . . . 4. ds : x:Id fp-> Type
. . . . . . 5. da : a:Knd fp-> Type
. . . . . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . . . . 7. w : World
. . . . . . 8. p : FairFifo
. . . . . . 9. FairFifo
. . . . . . 10. ∀i@0,x@0:Id.
. . . . . . |   vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . . . . |                             <ds,da,,,<k,x> : f,,,,·>
. . . . . . |                          else  fi)(x@0)?Top
. . . . . . 11. ∀i@0:Id, a:Action(i@0).
```

```
. . . . . . . |   ¬isnull(a)
. . . . . . . |   ⇒
. . . . . . . |   (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . . . . . |                                <ds,da,,,<k,x> : f,,,,·>
. . . . . . . |                             else  fi))(kind(a))?Top)
. . . . . . . 12. ∀l:IdLnk, tg:Id.
. . . . . . . |   (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . . . . . |                            <ds,da,,,<k,x> : f,,,,·>
. . . . . . . |                         else  fi))(rcv(l; tg))?Top
. . . . . . . 13. ∀i@0,x@0:Id.
. . . . . . . |   x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . . . . . |                       <ds,da,,,<k,x> : f,,,,·>
. . . . . . . |                    else  fi)))(x@0) ==> s(i@0;0).x@0
. . . . . . . |                                         =
. . . . . . . |                                         x0
. . . . . . . 14. ∀i@0:Id, t:ℕ.
. . . . . . . |   ¬isnull(a(i@0;t))
. . . . . . . |   ⇒
. . . . . . . |   ...
. . . . . . . |   & (∀x@0:Id.
. . . . . . . |      E != 1of(
. . . . . . . |           2of(2of(2of(2of(
. . . . . . . |           if eqof(IdDeq)(i@0,i)→
. . . . . . . |              <ds,da,,,<k,x> : f,,,,·>
. . . . . . . |           else  fi)))))(<kind(a(i@0;t))
. . . . . . . |                         ,x@0>) ==> s(i@0;t+1).x@0
. . . . . . . |                                    =
. . . . . . . |                                    E
. . . . . . . |                                    ((λx.s(i@0;t).x)
. . . . . . . |                                    ,val(a(i@0;t))))
. . . . . . . |   & (∀l:IdLnk.
. . . . . . . |      if eqof(IdDeq)(i@0,i)→ <ds,da,,,<k,x> : f,,,,·>
. . . . . . . |      else  fi.send(kind(a(i@0;t));l;λx.
. . . . . . . |      s(i@0;t).x;val(a(i@0;t));withlnk(l;m(i@0;t));i@0))
. . . . . . . |   & (∀x@0:Id.
. . . . . . . |      ¬L != 1of(
. . . . . . . |            2of(2of(2of(2of(2of(2of(
. . . . . . . |            if eqof(IdDeq)(i@0,i)→
. . . . . . . |               <ds,da,,,<k,x> : f,,,,·>
. . . . . . . |            else  fi)))))))(x@0) ==> ...
. . . . . . . |      ⇒
. . . . . . . |      s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . . . . . |   & (∀l:IdLnk, tg:Id.
. . . . . . . |      ¬L != 1of(
. . . . . . . |            2of(2of(2of(2of(2of(2of(2of(
. . . . . . . |            if eqof(IdDeq)(i@0,i)→
. . . . . . . |               <ds,da,,,<k,x> : f,,,,·>
. . . . . . . |            else  fi))))))))(<l,tg>) ==> ...
. . . . . . . |      ⇒
. . . . . . . |      w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . . . . . 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . . . . . 16. ∀e:E.
. . . . . . . |   loc(e) = i ⇒ (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . . . . . 17. j : Id
. . . . . . . 18. t : ℕ
```

```
. . . . . . . 19. valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . . . . . 20. j = i
. . . . . . . 21. kind(<j,t>) = k
. . . . . . . 22. ¬isnull(a(i;t))
. . . . . . . 23. islocal(kind(a(i;t)))
. . . . . . . |   ⇒
. . . . . . . |   P != (act(kind(a(i;t)))) ==> P
. . . . . . . |                                ((λx.s(i;t).x)
. . . . . . . |                                ,val(a(i;t)))
. . . . . . . 24. ∀x@0:Id.
. . . . . . . |   E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. . . . . . . |                                          =
. . . . . . . |                                          E
. . . . . . . |                                          ((λx.s(i;t).x)
. . . . . . . |                                          ,val(a(i;t)))
. . . . . . . 25. s(i;t+1).x = f((λx.s(i;t).x),val(a(i;t)))
. . . . . . . 26. z : Id
. . . . . . . 27. z = i
. . . . . . . ⊢  f((λz@0.s(z;t).z@0),val(a(z;t))) ∈ ds(x)?Top
. . . . . . . | by DoSubsume
. . . . . . . |
. . . . . . . |\
. . . . . . . | .....subterm .....
. . . . . . . | 1. i : Id
. . . . . . . | 2. x : Id
. . . . . . . | 3. k : Knd
. . . . . . . | 4. ds : x:Id fp-> Type
. . . . . . . | 5. da : a:Knd fp-> Type
. . . . . . . | 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . . . . . | 7. w : World
. . . . . . . | 8. p : FairFifo
. . . . . . . | 9. FairFifo
. . . . . . . | 10. ∀i@0,x@0:Id.
. . . . . . . |     vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . . . . . |                                <ds,da,,,<k,x> : f,,,,⋅>
. . . . . . . |                             else  fi)(x@0)?Top
. . . . . . . | 11. ∀i@0:Id, a:Action(i@0).
. . . . . . . |     ¬isnull(a)
. . . . . . . |     ⇒
. . . . . . . |     (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . . . . . |                                   <ds
. . . . . . . |                                   ,da
. . . . . . . |                                   ,
. . . . . . . |                                   ,
. . . . . . . |                                   ,<k,x> : f
. . . . . . . |                                   ,
. . . . . . . |                                   ,
. . . . . . . |                                   ,
. . . . . . . |                                   ,⋅>
. . . . . . . |                                else  fi))(kind(a))?Top)
. . . . . . . | 12. ∀l:IdLnk, tg:Id.
. . . . . . . |     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
. . . . . . . |                               <ds,da,,,<k,x> : f,,,,⋅>
. . . . . . . |                            else  fi))(rcv(l; tg))?Top
. . . . . . . | 13. ∀i@0,x@0:Id.
```

```
. . . . . . . |     x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . . . . . |                          <ds,da,,,<k,x> : f,,,,·>
. . . . . . . |                       else  fi)))(x@0) ==> s(i@0;0).x@0
. . . . . . . |                                             =
. . . . . . . |                                             x0
. . . . . . . | 14. ∀i@0:Id, t:ℕ.
. . . . . . . |     ¬isnull(a(i@0;t))
. . . . . . . |     ⇒
. . . . . . . |     (islocal(kind(a(i@0;t)))
. . . . . . . |      ⇒
. . . . . . . |      P != 1of(
. . . . . . . |           2of(2of(2of(
. . . . . . . |           if eqof(IdDeq)(i@0,i)→
. . . . . . . |              <ds,da,,,<k,x> : f,,,,·>
. . . . . . . |           else  fi))))(act(kind(a(i@0;t)))) ==> P
. . . . . . . |                                                  ((λx.
. . . . . . . |                                                    ...)
. . . . . . . |                                                  ,...))
. . . . . . . |     & (∀x@0:Id.
. . . . . . . |        E != 1of(
. . . . . . . |             2of(2of(2of(2of(
. . . . . . . |             if eqof(IdDeq)(i@0,i)→
. . . . . . . |                <ds,da,,,<k,x> : f,,,,·>
. . . . . . . |             else  fi)))))(<kind(a(i@0;t))
. . . . . . . |                            ,x@0>) ==> s(i@0;t+1).x@0
. . . . . . . |                                       =
. . . . . . . |                                       E
. . . . . . . |                                       ((λx.s(i@0;t).x)
. . . . . . . |                                       ,val(a(i@0;t))))
. . . . . . . |     & ...
. . . . . . . |     & (∀x@0:Id.
. . . . . . . |        ¬L != 1of(
. . . . . . . |              2of(2of(2of(2of(2of(2of(
. . . . . . . |              if eqof(IdDeq)(i@0,i)→
. . . . . . . |                 <ds,da,,,<k,x> : f,,,,·>
. . . . . . . |              else  fi)))))))(x@0) ==> ...
. . . . . . . |        ⇒
. . . . . . . |        s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . . . . . |     & (∀l:IdLnk, tg:Id.
. . . . . . . |        ¬L != 1of(
. . . . . . . |              2of(2of(2of(2of(2of(2of(2of(
. . . . . . . |              if eqof(IdDeq)(i@0,i)→
. . . . . . . |                 <ds,da,,,<k,x> : f,,,,·>
. . . . . . . |              else  fi))))))))(<l,tg>) ==> ...
. . . . . . . |        ⇒
. . . . . . . |        w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . . . . . | 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . . . . . | 16. ∀e:E.
. . . . . . . |     loc(e) = i
. . . . . . . |     ⇒
. . . . . . . |     (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . . . . . | 17. j : Id
. . . . . . . | 18. t : ℕ
. . . . . . . | 19. valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . . . . . | 20. j = i
```

```
. . . . . . . | 21. kind(<j,t>) = k
. . . . . . . | 22. ¬isnull(a(i;t))
. . . . . . . | 23. islocal(kind(a(i;t)))
. . . . . . . |     ⇒
. . . . . . . |     P != (act(kind(a(i;t)))) ==> P
. . . . . . . |                                   ((λx.s(i;t).x)
. . . . . . . |                                   ,val(a(i;t)))
. . . . . . . | 24. ∀x@0:Id.
. . . . . . . |     E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. . . . . . . |                                            =
. . . . . . . |                                            E
. . . . . . . |                                            ((λx.
. . . . . . . |                                              s(i;t).x)
. . . . . . . |                                            ,val(a(i;t)))
. . . . . . . | 25. s(i;t+1).x = f((λx.s(i;t).x),val(a(i;t)))
. . . . . . . | 26. z : Id
. . . . . . . | 27. z = i
. . . . . . . | ⊢  (λz@0.s(z;t).z@0) ∈ State(ds)
. . . . . . . |   by Unfold ‘ma-state‘ 0 THEN DoSubsume
. . . . . . . |      THEN
. . . . . . . |      HypSubst -2 0
. . . . . . . |      THEN
. . . . . . . |      BackThruSomeHyp
. . . . . . . |
. . . . . . . .\
. . . . . . . . .....subterm .....
. . . . . . . . 1. i : Id
. . . . . . . . 2. x : Id
. . . . . . . . 3. k : Knd
. . . . . . . . 4. ds : x:Id fp-> Type
. . . . . . . . 5. da : a:Knd fp-> Type
. . . . . . . . 6. f : State(ds)→ma-valtype(da; k)→ds(x)?Void
. . . . . . . . 7. w : World
. . . . . . . . 8. p : FairFifo
. . . . . . . . 9. FairFifo
. . . . . . . . 10. ∀i@0,x@0:Id.
. . . . . . . .     vartype(i@0;x@0) ⊆r 1of(if eqof(IdDeq)(i@0,i)→
. . . . . . . .                              <ds,da,,,<k,x> : f,,,,·>
. . . . . . . .                           else  fi)(x@0)?Top
. . . . . . . . 11. ∀i@0:Id, a:Action(i@0).
. . . . . . . .     ¬isnull(a)
. . . . . . . .     ⇒
. . . . . . . .     (valtype(i@0;a) ⊆r 1of(2of(if eqof(IdDeq)(i@0,i)→
. . . . . . . .                                 <ds
. . . . . . . .                                 ,da
. . . . . . . .                                 ,
. . . . . . . .                                 ,
. . . . . . . .                                 ,<k,x> : f
. . . . . . . .                                 ,
. . . . . . . .                                 ,
. . . . . . . .                                 ,
. . . . . . . .                                 ,·>

. . . . . . . .                              else  fi))(kind(a))?Top)
. . . . . . . . 12. ∀l:IdLnk, tg:Id.
. . . . . . . .     (w.M(l,tg)) ⊆r 1of(2of(if eqof(IdDeq)(source(l),i)→
```

```
. . . . . . . .                               <ds,da,,,<k,x> : f,,,,·>
. . . . . . . .                            else  fi))(rcv(l; tg))?Top
. . . . . . . . 13. ∀i@0,x@0:Id.
. . . . . . . .     x0 != 1of(2of(2of(if eqof(IdDeq)(i@0,i)→
. . . . . . . .                          <ds,da,,,<k,x> : f,,,,·>
. . . . . . . .                       else  fi)))(x@0) ==> s(i@0;0).x@0
. . . . . . . .                                            =
. . . . . . . .                                            x0
. . . . . . . . 14. ∀i@0:Id, t:ℕ.
. . . . . . . .     ¬isnull(a(i@0;t))
. . . . . . . .     ⇒
. . . . . . . .     (islocal(kind(a(i@0;t)))
. . . . . . . .      ⇒
. . . . . . . .      P != 1of(
. . . . . . . .           2of(2of(2of(
. . . . . . . .           if eqof(IdDeq)(i@0,i)→
. . . . . . . .              <ds,da,,,<k,x> : f,,,,·>
. . . . . . . .           else  fi))))(act(kind(a(i@0;t)))) ==> P
. . . . . . . .                                                 ((λx.
. . . . . . . .                                                   ...)
. . . . . . . .                                                 ,...))
. . . . . . . .     & (∀x@0:Id.
. . . . . . . .        E != 1of(
. . . . . . . .             2of(2of(2of(2of(
. . . . . . . .             if eqof(IdDeq)(i@0,i)→
. . . . . . . .                <ds,da,,,<k,x> : f,,,,·>
. . . . . . . .             else  fi)))))(<kind(a(i@0;t))
. . . . . . . .                           ,x@0>) ==> s(i@0;t+1).x@0
. . . . . . . .                                      =
. . . . . . . .                                      E
. . . . . . . .                                      ((λx.s(i@0;t).x)
. . . . . . . .                                      ,val(a(i@0;t))))
. . . . . . . .     & ...
. . . . . . . .     & (∀x@0:Id.
. . . . . . . .        ¬L != 1of(
. . . . . . . .              2of(2of(2of(2of(2of(2of(
. . . . . . . .              if eqof(IdDeq)(i@0,i)→
. . . . . . . .                 <ds,da,,,<k,x> : f,,,,·>
. . . . . . . .              else  fi)))))))(x@0) ==> ...
. . . . . . . .        ⇒
. . . . . . . .        s(i@0;t).x@0 = s(i@0;t+1).x@0)
. . . . . . . .     & (∀l:IdLnk, tg:Id.
. . . . . . . .        ¬L != 1of(
. . . . . . . .              2of(2of(2of(2of(2of(2of(2of(
. . . . . . . .              if eqof(IdDeq)(i@0,i)→
. . . . . . . .                 <ds,da,,,<k,x> : f,,,,·>
. . . . . . . .              else  fi))))))))(<l,tg>) ==> ...
. . . . . . . .        ⇒
. . . . . . . .        w-tagged(tg; onlnk(l;m(i@0;t))) = nil)
. . . . . . . . 15. ∀x:Id. vartype(i;x) ⊆r ds(x)?Top
. . . . . . . . 16. ∀e:E.
. . . . . . . .     loc(e) = i
. . . . . . . .     ⇒
. . . . . . . .     (valtype(e) ⊆r ma-valtype(da; kind(e)))
. . . . . . . . 17. j : Id
```

```
. . . . . . . . 18. t : ℕ
. . . . . . . . 19. valtype(i;a(i;t)) ⊆r ma-valtype(da; k)
. . . . . . . . 20. j = i
. . . . . . . . 21. kind(<j,t>) = k
. . . . . . . . 22. ¬isnull(a(i;t))
. . . . . . . . 23. islocal(kind(a(i;t)))
. . . . . . . .     ⇒
. . . . . . . .     P != (act(kind(a(i;t)))) ==> P
. . . . . . . .                                  ((λx.s(i;t).x)
. . . . . . . .                                  ,val(a(i;t)))
. . . . . . . . 24. ∀x@0:Id.
. . . . . . . .     E != <k,x> : f(<kind(a(i;t)),x@0>) ==> s(i;t+1).x@0
. . . . . . . .                                             =
. . . . . . . .                                             E
. . . . . . . .                                             ((λx.
. . . . . . . .                                               s(i;t).x)
. . . . . . . .                                             ,val(a(i;t)))
. . . . . . . . 25. s(i;t+1).x = f((λx.s(i;t).x),val(a(i;t)))
. . . . . . . . 26. z : Id
. . . . . . . . 27. z = i
. . . . . . . . ⊢  val(a(z;t)) ∈ ma-valtype(da; k)
. . . . . . . .   by DoSubsume THEN StrongHypSubst -1 0
END
```