d-comp[11,40]

```
d-comp(D; v; sched; dec; d)(t,f)
≡def let s= λi.if (t =0 0) then λx.M(i).init(x)?v(i,x) else (f((t - 1),i)).1 fi in
        λi.
       let si= s(i) in
           let w= d-partial-world(D;f;t;s;d) in
               let a = case sched(i)
                 of inl(f) => case f(t)
                           of inl(l) => if destination(l) = i
                                          ∧b 0 <z ‖queue(l;t)‖
                                        then doact(rcv(l,mtag(hd(queue(l;t))))
                                                ;mval(hd(queue(l;t))))
                                        else null
                                        fi
                            | inr(a) => if isl(dec
                                            (i
                                            ,a
                                            ,w-knum(w;i;locl(a);t)
                                            ,read-state(si)))
                                        then doact(inr a
                                                ;outl(dec
                                                    (i
                                                    ,a
                                                    ,w-knum(w;i;locl(a);t)
                                                    ,read-state(si))))
                                        else null
                                        fi
                  | inr(x) => null in
                  let m = if isl(a)
                     then []
                     else filter(λm.source(mlnk(m)) = i
                           ;M(i).sends(outr(a).1,read-state(si),outr(a).2))
                     fi in
                      let s′ = if isl(a)
                        then si
                        else λx.M(i).ef(outr(a).1,x,read-state(si),outr(a).2)
                                 ?si(x)
                        fi in <shift-state(s′), a, m>
```

clarification:

```
d-comp(D; v; sched; dec; d)(t,f)
≡def let s= λi.if (t =0 0) then λx.d-m(D; i).init(x)?v(i,x) else (f((t - 1),i)).1 fi in
```

```
 λi.
let si= s(i) in
      let w= d-partial-world(D;f;t;s;d) in
            let a = case sched(i)
                of inl(f) => case f(t)
                                of inl(l) => if destination(l) = i
                                               ∧b 0 <z ||w-queue(w; l; t)||
                                             then doact(rcv(l,mtag(hd(w-queue
                                                          (w; l; t))))
                                                          ;mval(hd(w-queue(w; l; t))))
                                             else null
                                             fi
                                 | inr(a) => if isl(dec
                                                    (i
                                                    ,a
                                                    ,w-knum(w;i;locl(a);t)
                                                    ,read-state(si)))
                                              then doact(inr a
                                                          ;outl(dec
                                                                 (i
                                                                 ,a
                                                                 ,w-knum(w;i;locl(a);t)
                                                                 ,read-state(si))))
                                              else null
                                              fi
                 | inr(x) => null in
                 let m = if isl(a)
                     then []
                     else filter(λm.source(mlnk(m)) = i
                                ;d-m(D; i).sends(outr(a).1
                                                   ,read-state(si)
                                                   ,outr(a).2))
                     fi in
                      let s′ = if isl(a)
                          then si
                          else λx.d-m(D; i).ef(outr(a).1
                                                 ,x
                                                 ,read-state(si)
                                                 ,outr(a).2)
                                                ?si(x)
                          fi in <shift-state(s′), a, m>
```