R-Dsys^{0,22}

[[R]]
$\equiv_{\text{def}}$ case R of
Rnone =>
Rplus(*left*,*right*)=>rec_1,rec_2.$rec_1 \oplus rec_2$
Rinit(loc,T,x,v)=> @loc: x:T
initially $x = v$
Rframe(loc,T,x,L)=> @loc: only L affects $x : T$
Rsframe(lnk,tag,L)=> @source(lnk): only L sends on (lnk with tag)
Reffect(loc,ds,knd,T,x,f)=> @loc: with declarations
ds:ds
da:knd : T

effect of knd(v) is $x := f$ s v
Rsends(ds,knd,T,l,dt,g)=> @source(l): with declarations
ds:ds
da:knd : $T \oplus$ lnk-decl(l;dt)
knd(v) sends g s v on link l
Rpre(loc,ds,a,T,P)=> @loc (with ds: ds
action a:T
precondition a(v) is
P s v)
Raframe(loc,k,L)=> @loc: k affects only members of L
Rbframe(loc,k,L)=> @loc: k sends only links in L
Rrframe(loc,x,L)=> @loc: only members of L read x

clarification:

[[R]]
$\equiv_{\text{def}}$ case R of
Rnone =>
Rplus(*left*,*right*)=>rec_1,rec_2.$rec_1 \oplus rec_2$
Rinit(loc,T,x,v)=> @loc: x:T
initially $x = v$
Rframe(loc,T,x,L)=> @loc: only L affects $x : T$
Rsframe(lnk,tag,L)=> @source(lnk): only L sends on (lnk with tag)
Reffect(loc,ds,knd,T,x,f)=> @loc: with declarations
ds:ds
da:knd : T

effect of knd(v) is $x := f$ s v
Rsends(ds,knd,T,l,dt,g)=> @source(l): with declarations
ds:ds

da:fpf-join(KindDeq;knd : T;lnk-decl(l;dt))
 knd(v) sends g s v on link l
Rpre(loc,ds,a,T,P)=> @loc (with ds: ds
 action a:T
 precondition a(v) is
 P s v)
Raframe(loc,k,L)=> @loc: k affects only members of L
Rbframe(loc,k,L)=> @loc: k sends only links in L
Rrframe(loc,x,L)=> @loc: only members of L read x

http://www.cs.cornell.edu/Info/Projects/NuPrl/FDLcontent/p0_286125_/p35_146209_{R-Dsys}.html