

R-compat^{11,40}

```
R-compat{i:l}
  (A; B)
≡def ifRplus?(A)
  then R-compat{i:l}(Rplus-left(A); B) ∧ R-compat{i:l}(Rplus-right(A); B)
ifRplus?(B)
  then R-compat{i:l}(A; Rplus-left(B)) ∧ R-compat{i:l}(A; Rplus-right(B))
ifRnone?(A)
  then True
ifRnone?(B)
  then True
ifeq_id(R-loc(A); R-loc(B))
  then (fpf-compatible(Id; x.Type; id-deq; Rds(A); Rds(B))
    ∧ fpf-compatible(Knd; x.Type; Kind-deq; Rda(A); Rda(B)))
    ∧ if eq_bd(R-base-domain(A); R-base-domain(B))
      then A = B
      else R-frame-compat(A; B) ∧ R-frame-compat(B; A) ∧ R-discrete_compat(A; B)
    fi
  else R-interface-compat(A; B) ∧ R-interface-compat(B; A)
fi
```

clarification:

```
R-compat{i:l}
  (A; B)
≡def ifRplus?(A)
  then R-compat{i:l}(Rplus-left(A); B) ∧ R-compat{i:l}(Rplus-right(A); B)
ifRplus?(B)
  then R-compat{i:l}(A; Rplus-left(B)) ∧ R-compat{i:l}(A; Rplus-right(B))
ifRnone?(A)
  then True
ifRnone?(B)
  then True
ifeq_id(R-loc(A); R-loc(B))
  then (fpf-compatible(Id; x.Type{i}; id-deq; Rds(A); Rds(B))
    ∧ fpf-compatible(Knd; x.Type{i}; Kind-deq; Rda(A); Rda(B)))
    ∧ if eq_bd(R-base-domain(A); R-base-domain(B))
      then A = B ∈ es_realizer{i:l}
      else R-frame-compat(A; B) ∧ R-frame-compat(B; A) ∧ R-discrete_compat(A; B)
    fi
  else R-interface-compat(A; B) ∧ R-interface-compat(B; A)
fi
```

(recursive)

