Programming
Weak Synchronization
Models

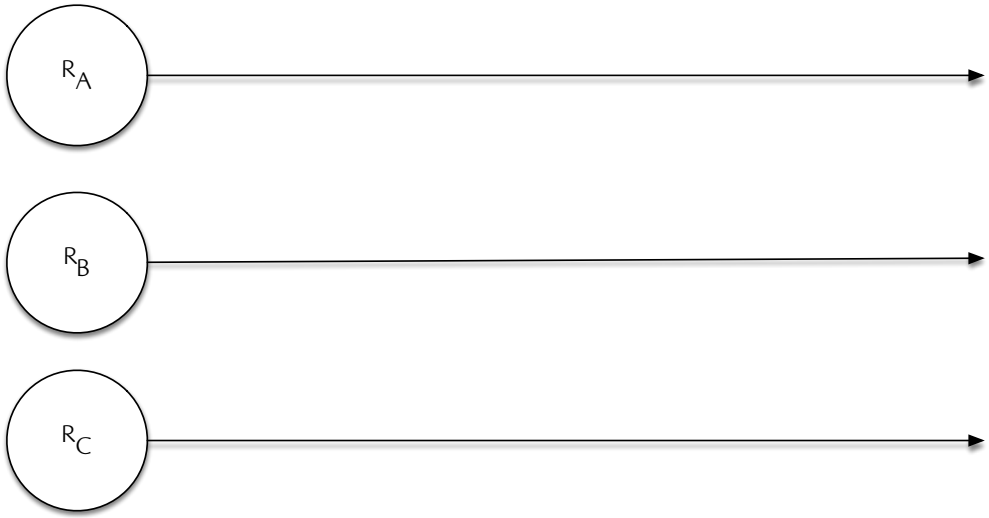
Christopher S. Meiklejohn
Université catholique de Louvain, Belgium
Instituto Superior Técnico, Portugal

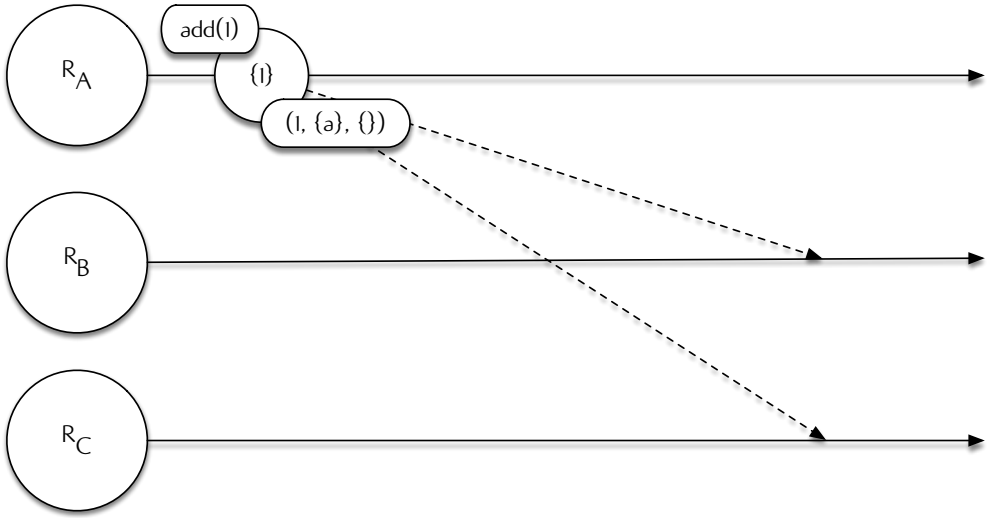
Convergent Objects
Conflict-Free
Replicated Data Types

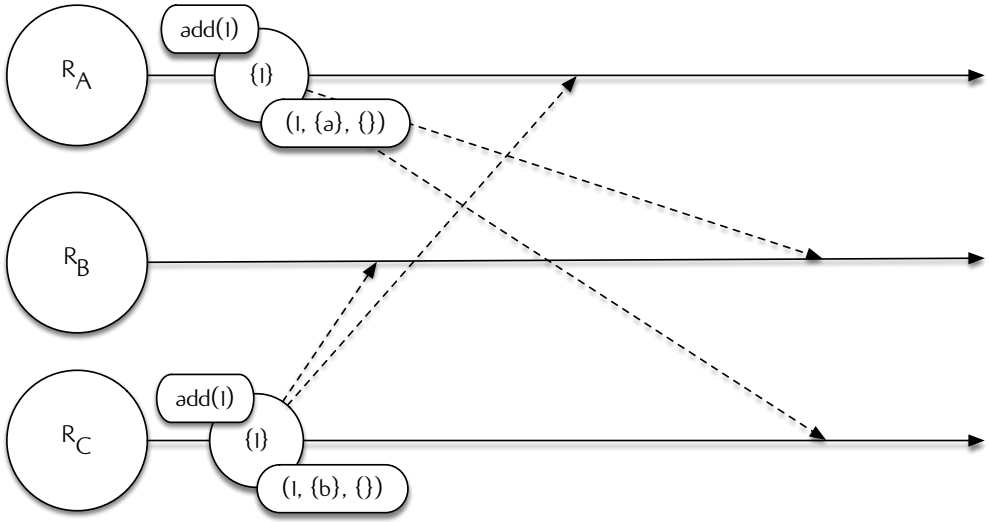
Conflict-Free Replicated Data Types

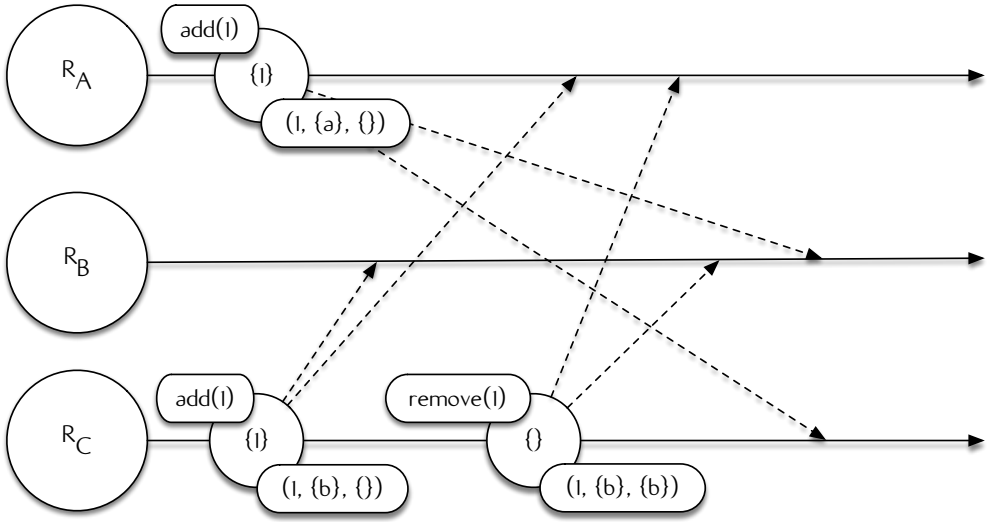
- **Many types exist with different properties**
  Sets, counters, registers, flags, maps
- **Strong Eventual Consistency**
  Instances satisfy SEC property per-object
- **Bounded join-semilattices**
  Formalized using bounded join-semilattices where the merge operation is the **join**

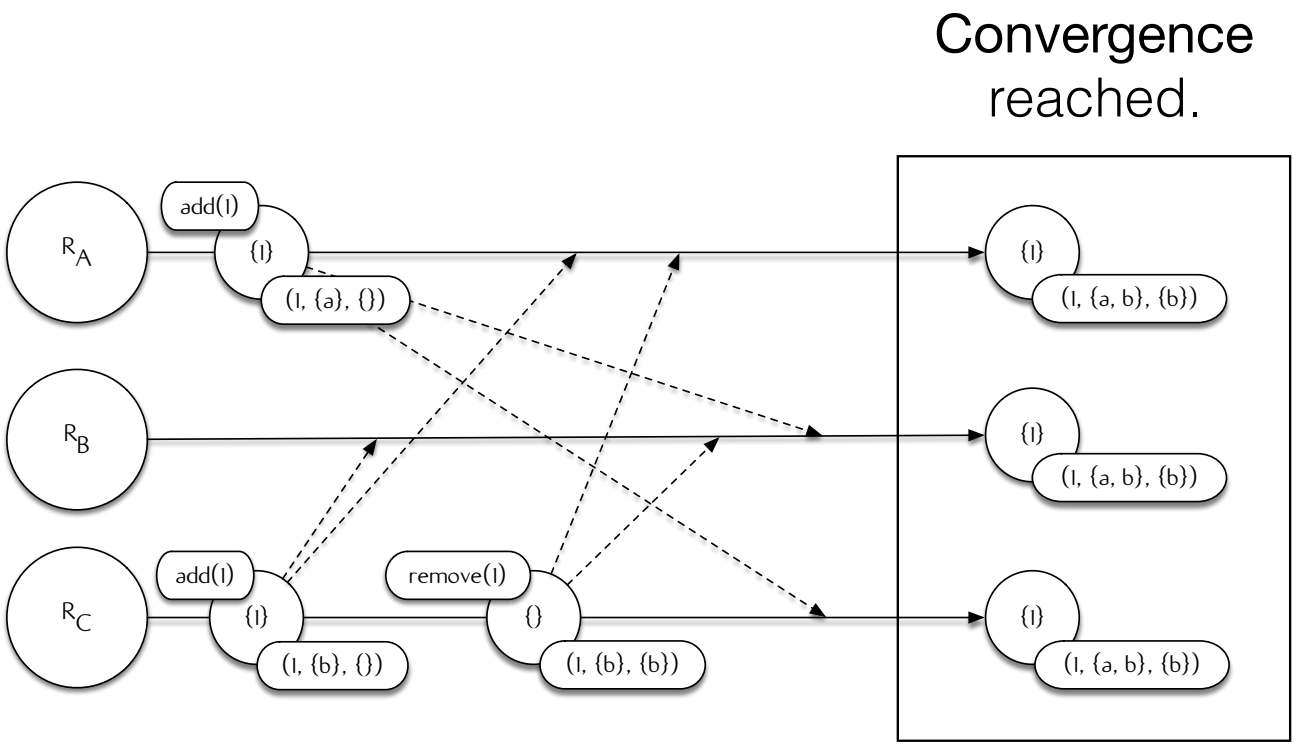
Convergent Objects
Observed-Remove Set


R_A
R_B
R_C

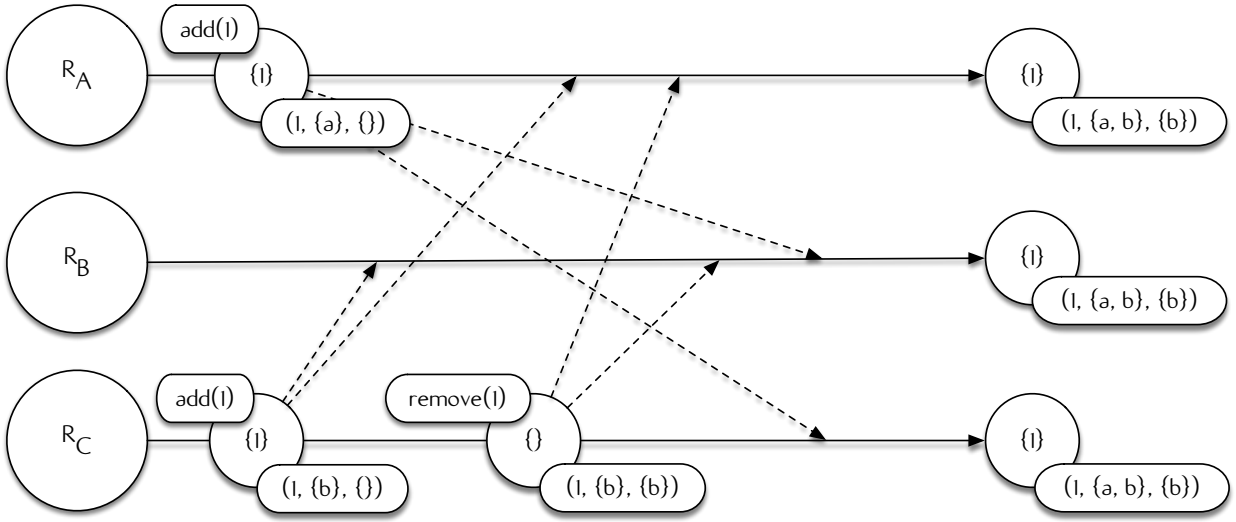

add(1)
R_A
{1}
(1, {a}, {})
R_B
R_C


add(1)
R_A
{1}
(1, {a}, {})
R_B
add(1)
R_C
{1}
(1, {b}, {})

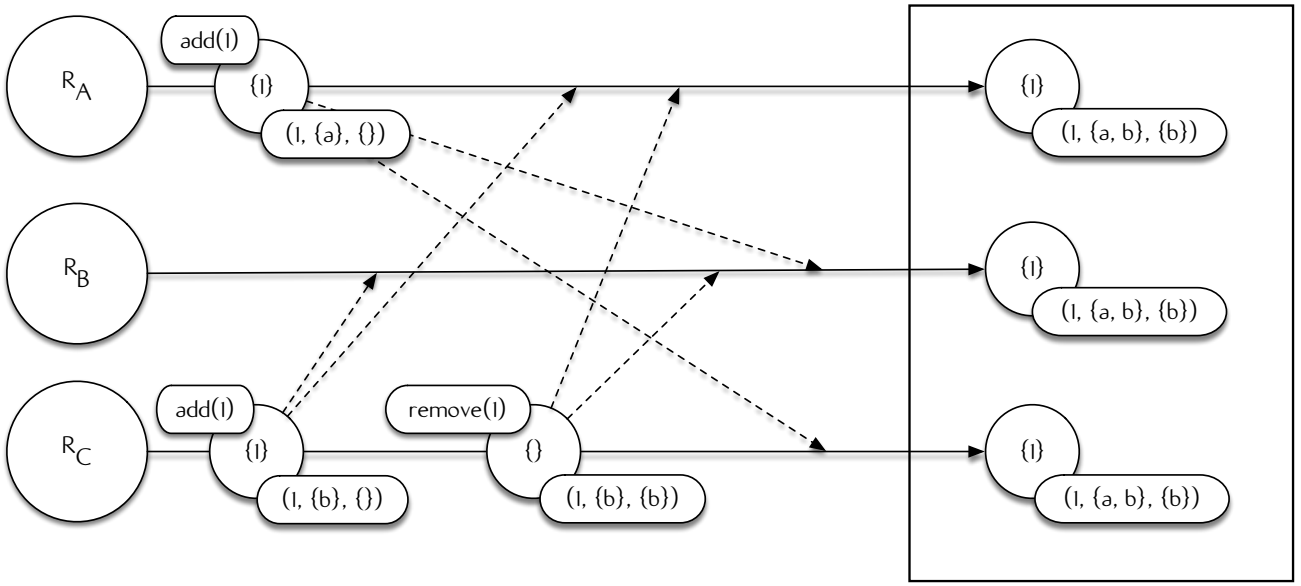

R_A
add(1)
{1}
(1, {a}, {})
R_B
R_C
add(1)
{1}
(1, {b}, {})
remove(1)
{}
(1, {b}, {b})


Convergence
reached.
R_A
R_B
R_C
add(1)
{1}
(1, {a}, {})
add(1)
{1}
(1, {b}, {})
remove(1)
{}
(1, {b}, {b})
{1}
(1, {a, b}, {b})
{1}
(1, {a, b}, {b})
{1}
(1, {a, b}, {b})

Convergent Objects
Nondeterminism

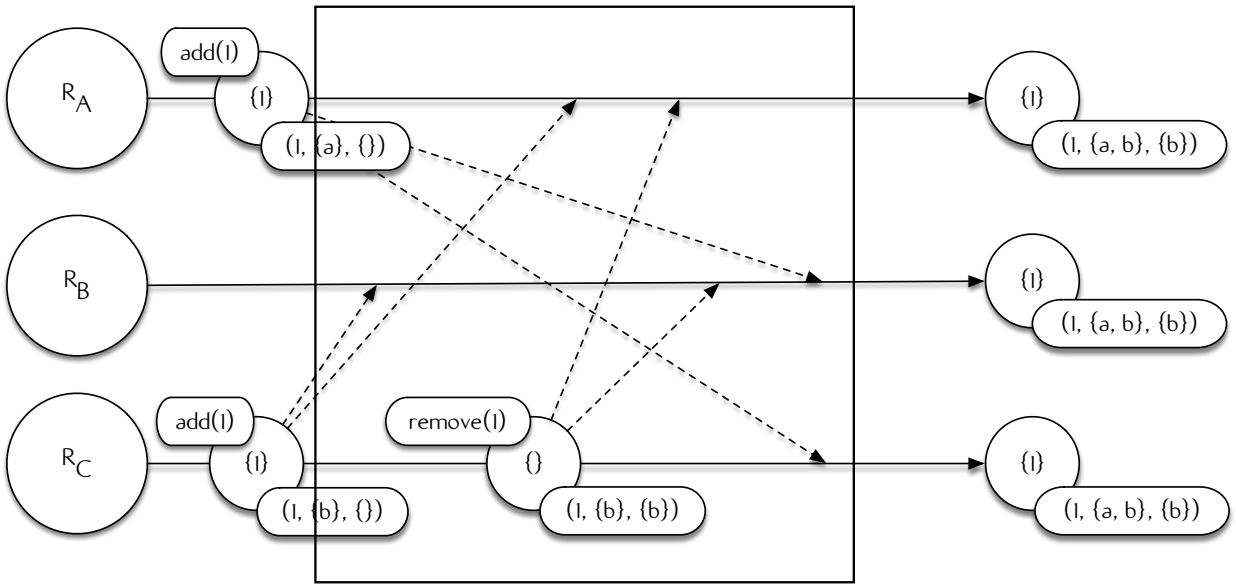

R_A
add(1)
{1}
(1, {a}, {})
R_B
R_C
add(1)
{1}
(1, {b}, {})
remove(1)
{}
(1, {b}, {b})
{1}
(1, {a, b}, {b})
{1}
(1, {a, b}, {b})
{1}
(1, {a, b}, {b})

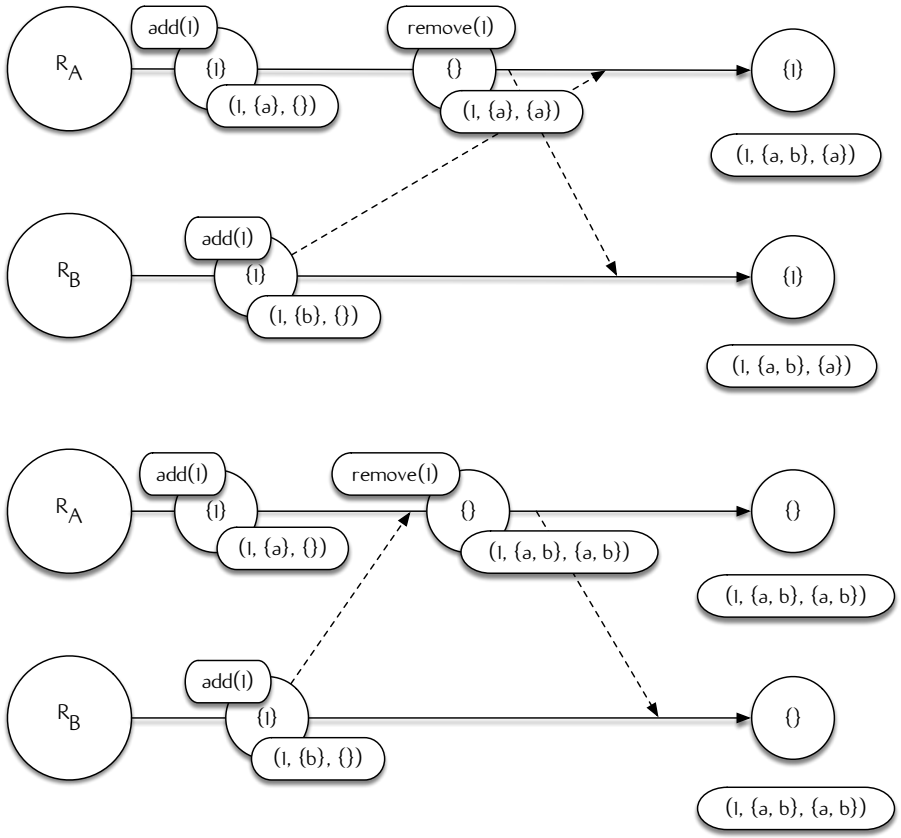
Convergence
reached.

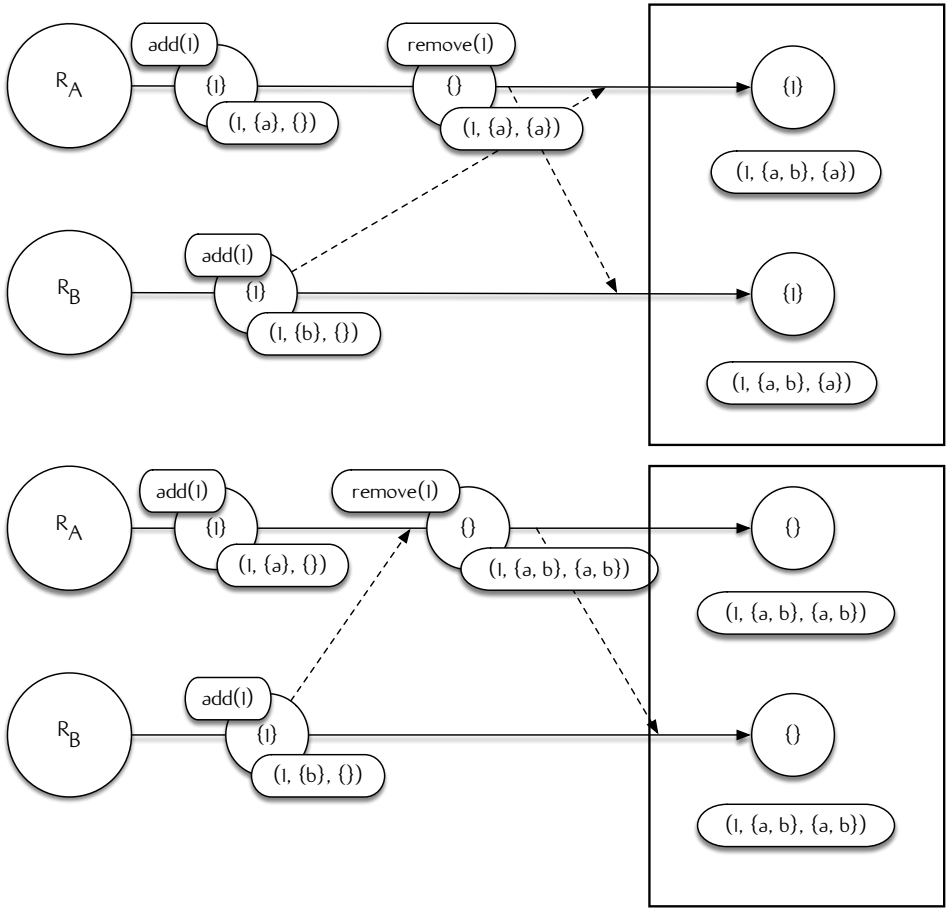


Desire:

The ability to reorder messages without impacting outcome.

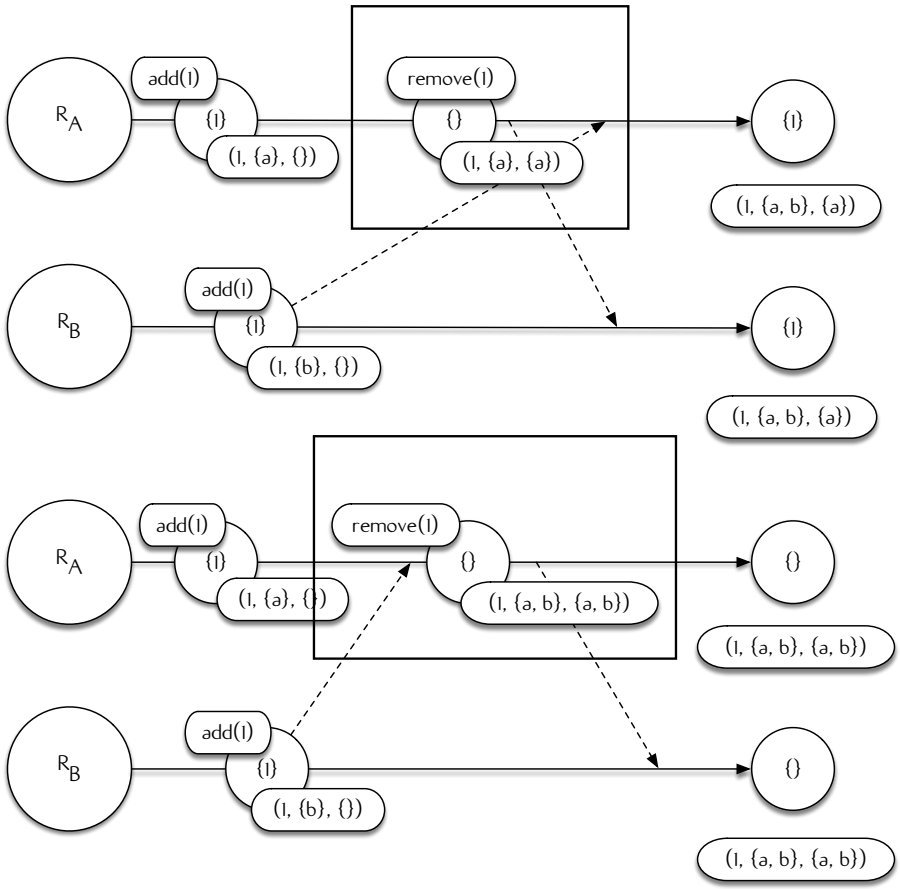



R_A
add(1)
{1}
(1, {a}, {})
remove(1)
{}
(1, {a}, {a})
{1}
(1, {a, b}, {a})
R_B
add(1)
{1}
(1, {b}, {})
{1}
(1, {a, b}, {a})
R_A
add(1)
{1}
(1, {a}, {})
remove(1)
{}
(1, {a, b}, {a, b})
{}
(1, {a, b}, {a, b})
R_B
add(1)
{1}
(1, {b}, {})
{}
(1, {a, b}, {a, b})



Convergence reached.

Different **synchronization schedules** can reach different outcomes.

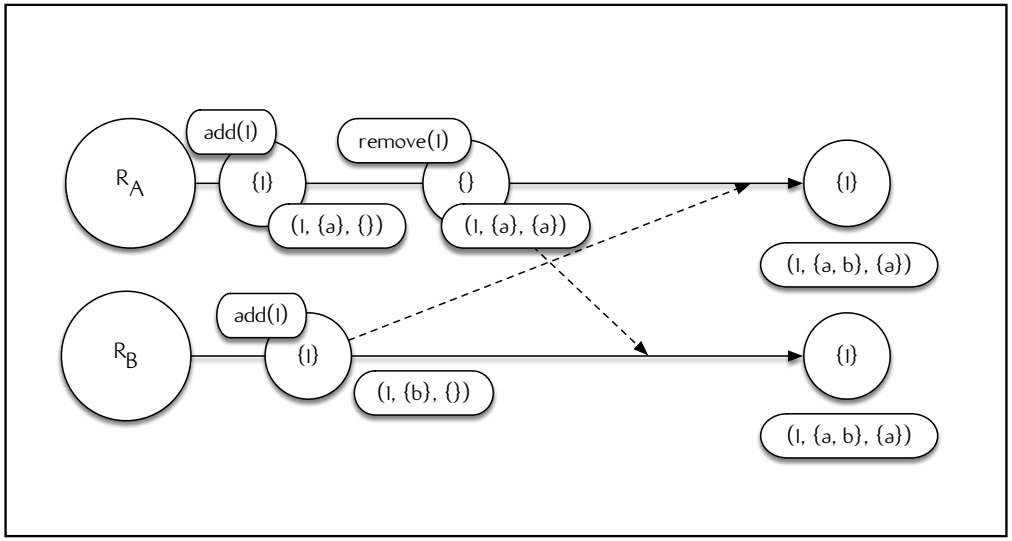


Reordering must be compatible with **causality.**

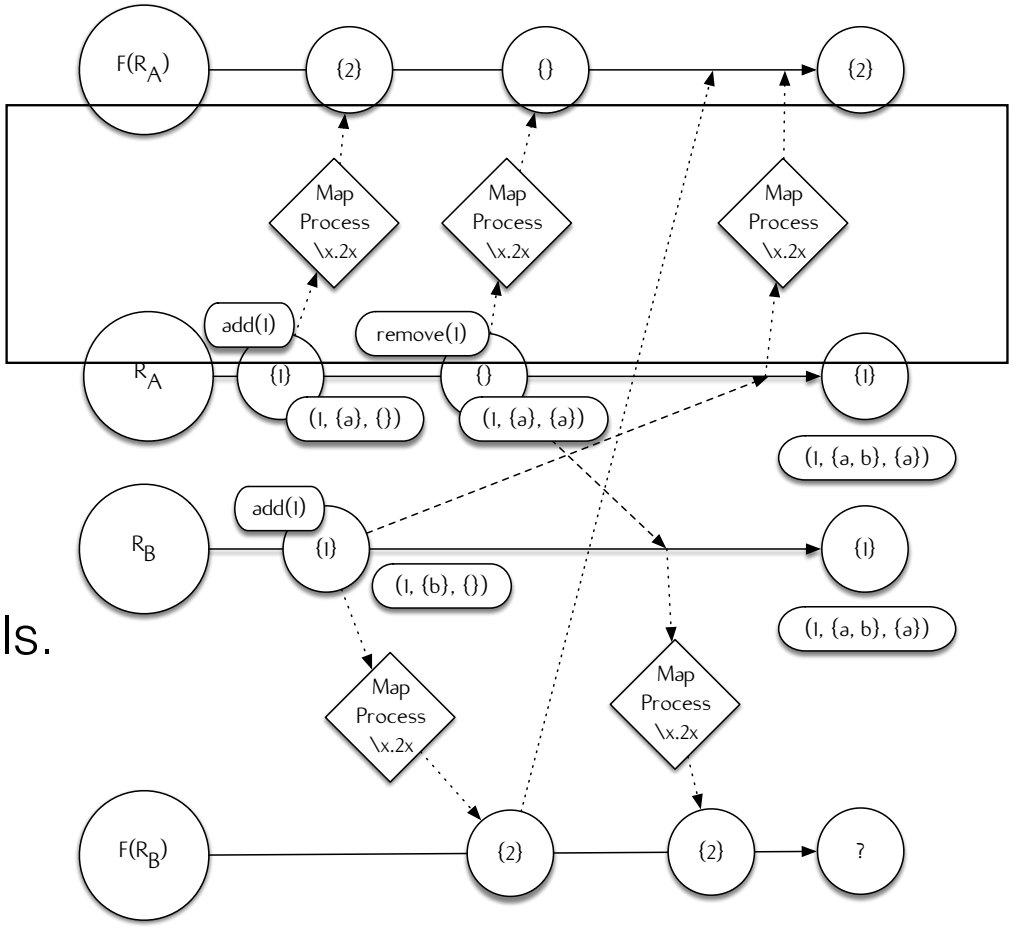
Each of these removes differ by their **causal** "influences."

Convergent Objects
Composition

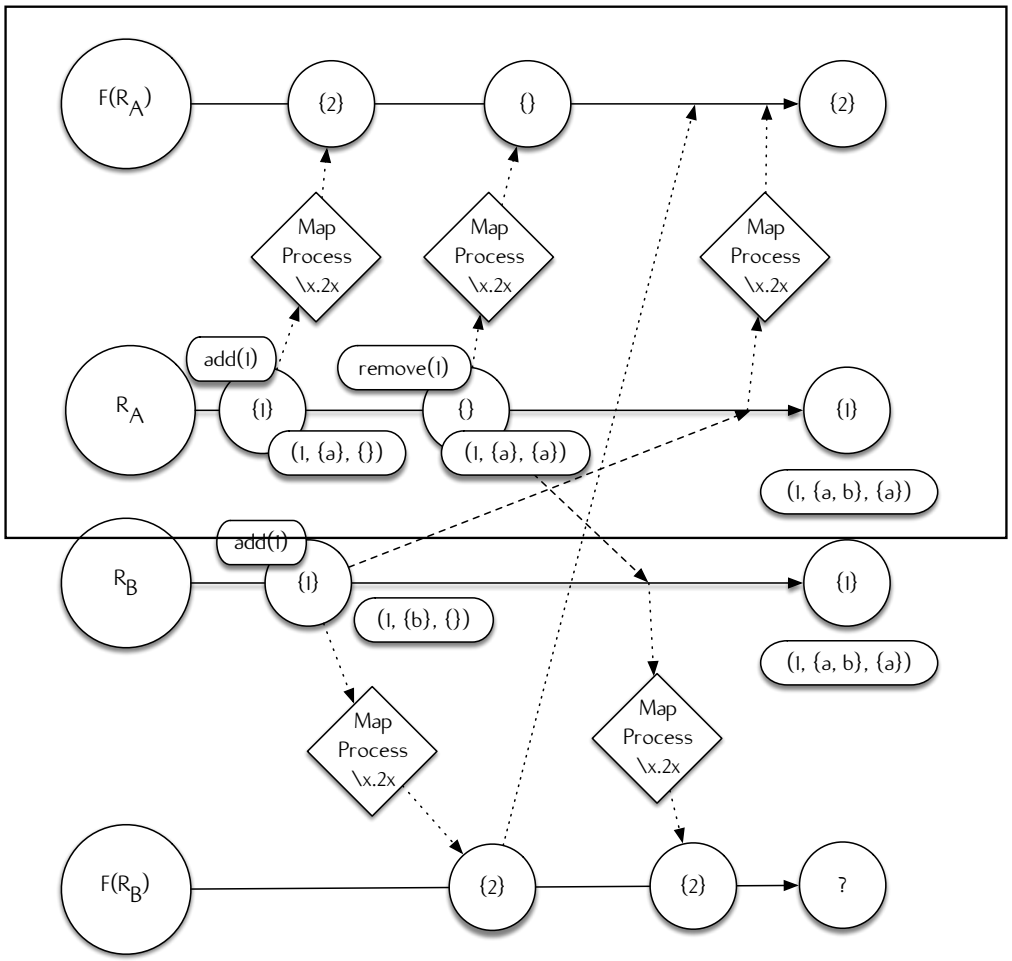
Replicated
set of naturals
across two nodes.

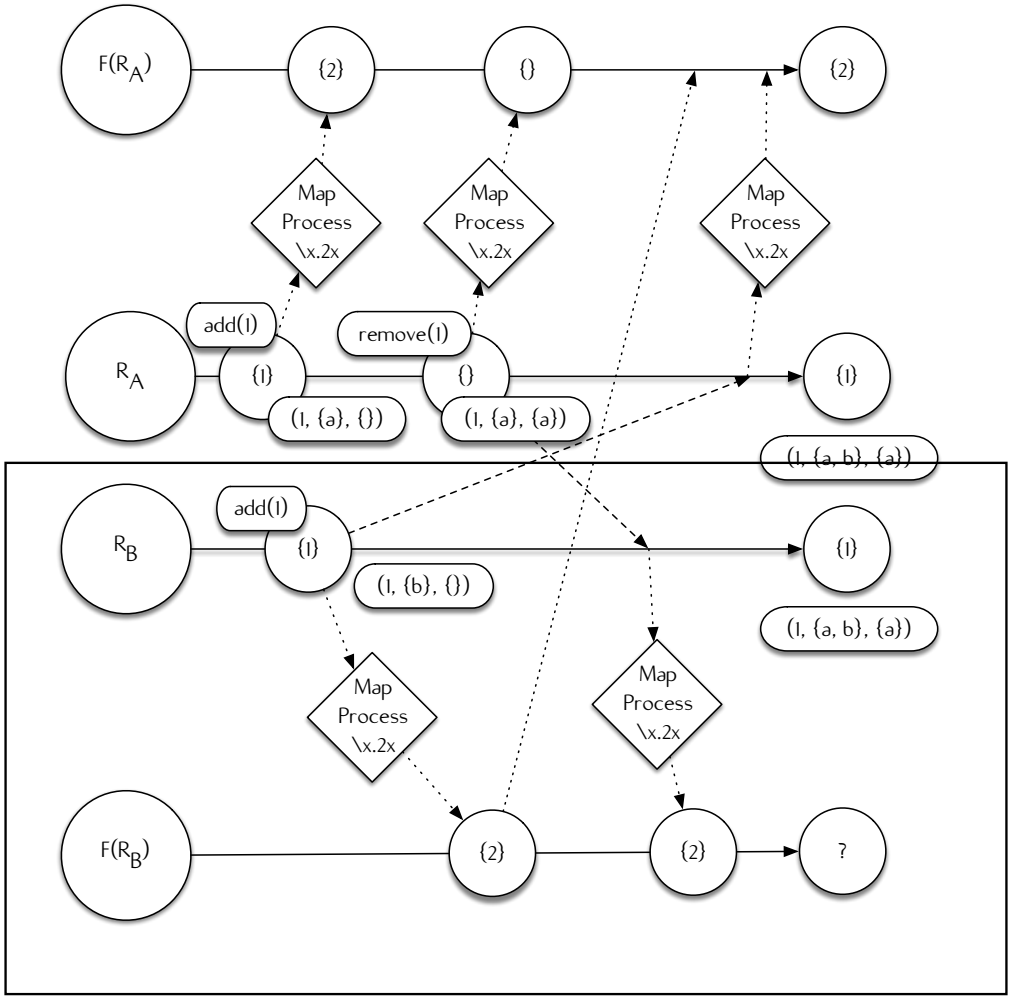


Map \x.2x
over a set of naturals.

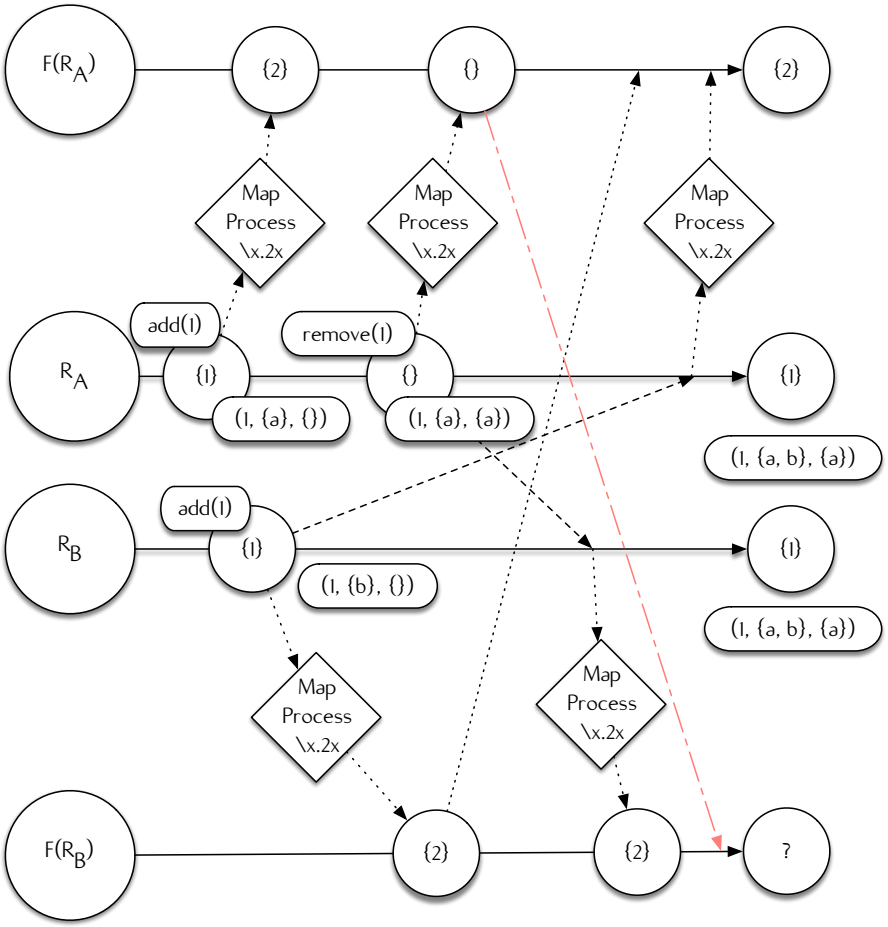


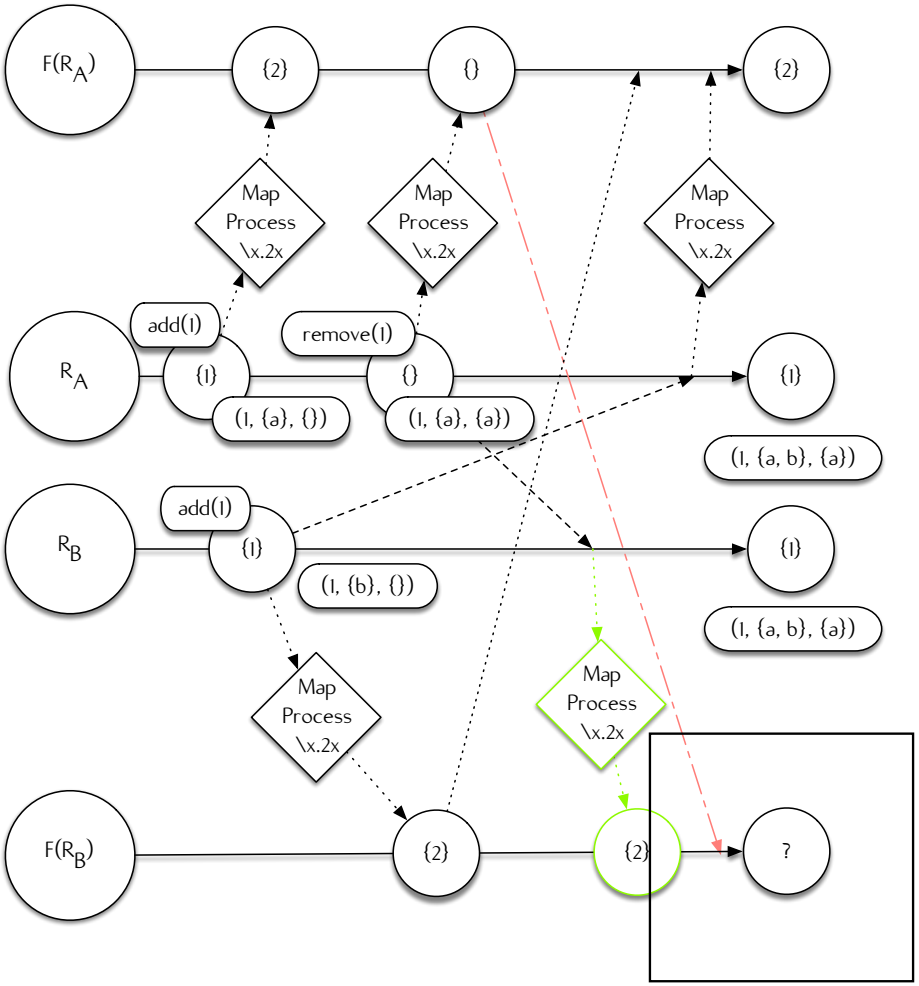
One **node**…


F(R_A)
{2}
{}
{2}
Map Process \x.2x
Map Process \x.2x
Map Process \x.2x
add(1)
remove(1)
R_A
{1}
{}
{1}
(1, {a}, {})
(1, {a}, {a})
(1, {a, b}, {a})
add(1)
R_B
{1}
{1}
(1, {b}, {})
(1, {a, b}, {a})
Map Process \x.2x
Map Process \x.2x
F(R_B)
{2}
{2}
?

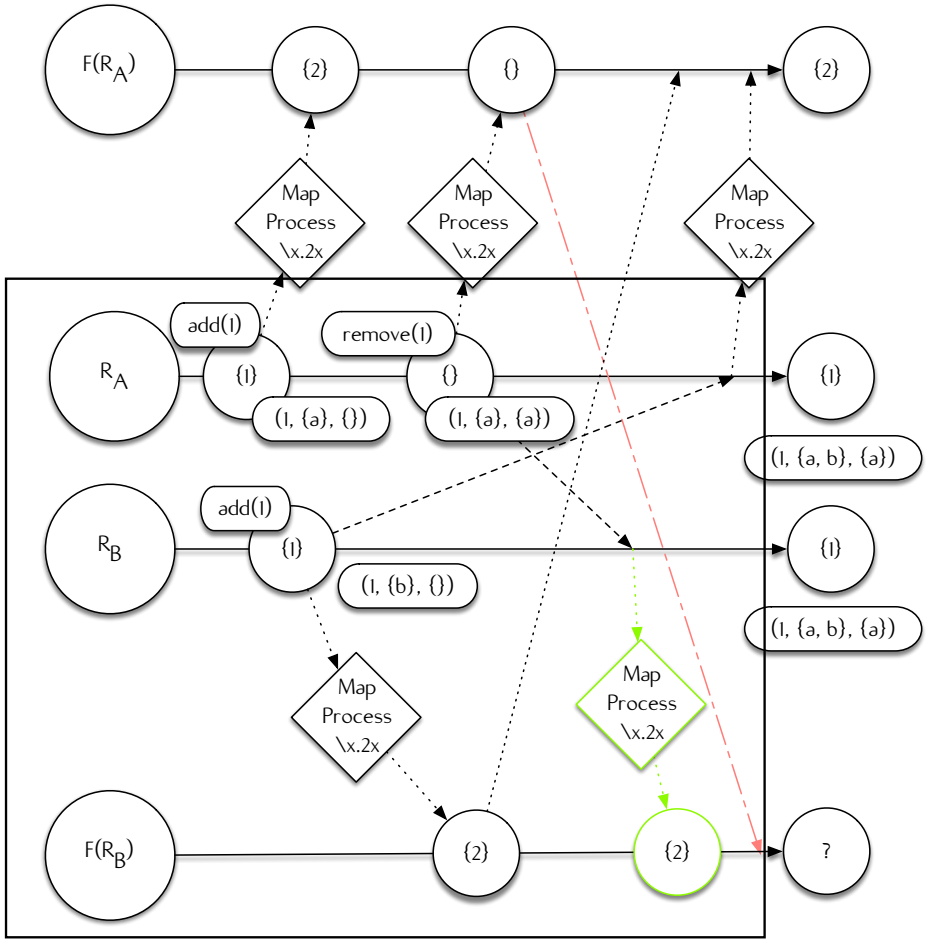


…another **node**.

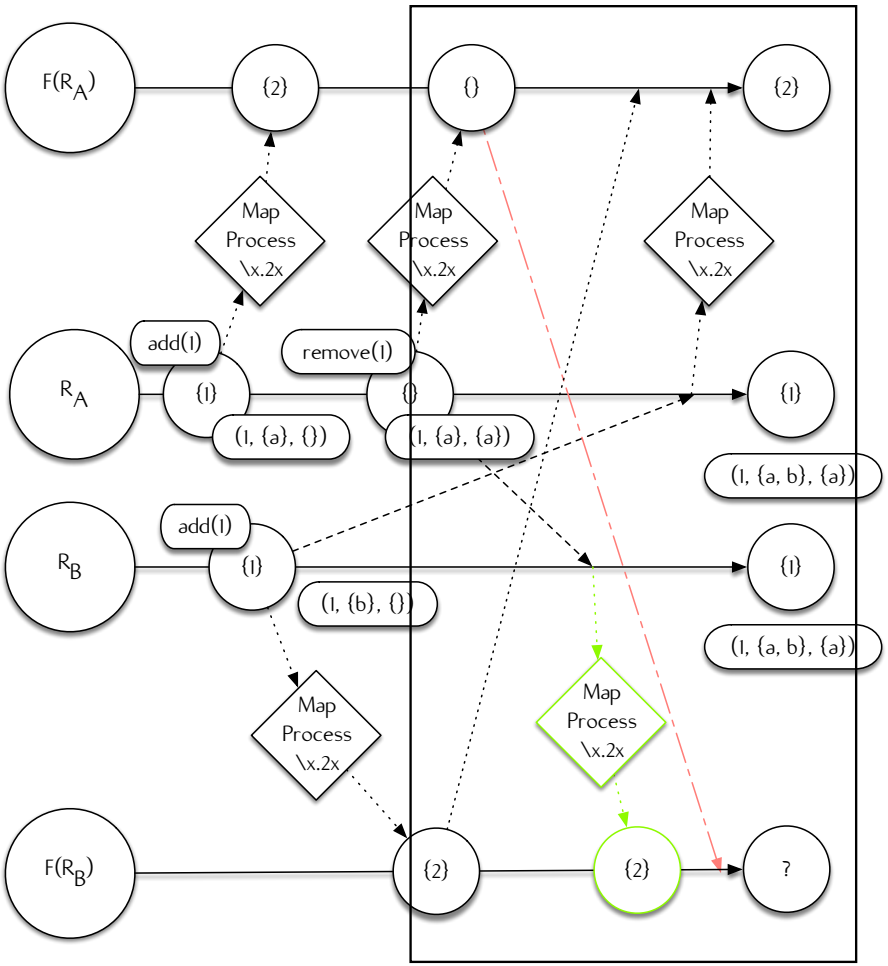

F(R_A)
{2}
{}
{2}
Map Process \x.2x
Map Process \x.2x
Map Process \x.2x
R_A
add(1)
remove(1)
{1}
{}
{1}
(1, {a}, {})
(1, {a}, {a})
(1, {a, b}, {a})
R_B
add(1)
{1}
{1}
(1, {b}, {})
(1, {a, b}, {a})
Map Process \x.2x
Map Process \x.2x
F(R_B)
{2}
{2}
?



Nondeterministic outcome.



Correct output that's seen all updates.



"Earlier" value that's been **delayed.**