State-based CRDTs

A tuple ($\mathbf{S}$, $\mathbf{s^0}$, $\mathbf{q}$, $\mathbf{u}$, $\mathbf{m}$) where

- $\mathbb{S}$ is the set of possible states for the CRDT

- $s0$ is the initial state of the CRDT

- q is the *query* function that lets a client read the current state of the object

- u is the *update* function that lets a client alter the state of the object

- m is a binary *merge* function

More formally

State-based CRDTs

More formally

A tuple ($\mathbf{S}$, $\mathbf{s0}$, $\mathbf{q}$, $\mathbf{u}$, $\mathbf{m}$) where

- S is the set of possible states for the CRDT

- $\mathrm{s0}$ is the initial state of the CRDT

- q is the *query* function that lets a client read the current state of the object

- u is the *update* function that lets a client alter the state of the object

- m is a binary *merge* function

State-based CRDTs

More formally

m needs to be **associative**

$$\forall x, y, z : merge(x, (merge(y, z))) = merge(merge(x, y), z)$$