Conflict-Free Replicated Data Types

Lattice Theory for Parallel Programming

Goals

- ✯ Working with the formal definitions of CRDTs.
- ✯ Implementing various CRDTs.

Exercise 1 – Formal CRDT

The following exercices are also available in the slides of the course (useful for more context).

1. Using lexicographic order, design a CRDT implementing a PN-Counter.
2. Find a proper way to compare the behaviors of the three CRDTs implementing a PN-Counter.
   Prove that all definitions are equivalent.
3. In the G-Set CRDT, prove that $lookup_x \triangleq \lambda S. lookup(S, x)$ is a monotone function. Recall:
   $$\begin{aligned} &lookup : \mathcal{P}(X) \times X \to \mathbb{B} \\ &lookup(S, x) \triangleq x \in S \end{aligned}$$
4. Define the 2P-Set CRDT using GS and the decreasing-only set CRDT DS.
5. For the OR-Set CRDT, prove the order $\leq$ and the join $\sqcup$ operations are coherent: $X \leq Y \Leftrightarrow X \sqcup Y = Y$.
6. Define the OR-Set CRDT without having to redefine yourself the lattice operations.

Exercise 2 – OR-Set CRDT

1. Implement and test the OR-Set CRDT as seen in class, extending your implementation of the G-Counter.
2. Implement an operation-based CRDT of the OR-Set.