Reading notes from branch 5-export-import-crdt-to-from-json-string | Marc Shapiro, 21 May 2020

Separation of metadata from data: does it work for nested structures? Let's say we have A (maybe a map) that contains B (say, an RGA) and C (say, a custom lattice). One way to do it is:

```
A = {
    A.metadata
    A.data  = {
        B = { B.metadata; B.data}
        C = { C.metadata; C.data}
}
```

But what we really want is

```
A = {
    A.metadata = {consolidated_VV; A.specific_metadata; B.metadata; C.metadata }
    A.data = {A.specific_data; B.data; C.data }
}
```

Which one do you implement?

I went quickly over the Kotlinx serialisation documentation. "JSON Transformations" looks like a good way todo it.

I don't understand the logic of delta generation; please explain it to me some day.

DCIdTest.kt

The name is probably misleading. What is a Dcid? Why compare them? why are they ordered???

In **DCId.kt:** "This class represents a datacenter (DC) identifier (id)" = then it would be immutable and not ordered.

TimestampTest.kt

Need to plan for wrap-around of timestamps! or eliminate somehow.

VersionVectorTest.kt

todo: $vc1 || vc2 <==> ¬(vc1<vc2) ∧ ¬(vc2 < vc1)$

todo: $¬ (vc1 < vc2) ≠ vc1 ≥= vc2$ + variations

wrap-around?

To/from json too simplistic; test more complex cases

Do VVs support the dynamic adding or removing of a DC?

ImmutableTest.kt

Replace "immutable" type with "customSemiLattice" type.

Is there any way to test that something immutable is never ever changed???

PNCounterTest.kt

test to/from JSON = comparing string — very low level, is there a better approach?

Missing: version number in JSON

LWWRegisterTest.kt

Same comments about JSON.

The interface seems awfully complex (environment, timestamp); over-engineered?

MVRegisterTest.kt

I don't think we should allow user-level operations on a multi-branch MVR, other than reading and testing (merge is OK). Other kinds of operations should be disallowed unless the user first assigns a single value.

assignAssignOldGet() "evaluates the scenario: assign assign(old timestamp) get. Call to get should return first assigned value." So an MVR is also a LWW? Over-engineered; simplify to standard register on sequential assignments.

Same comments about JSON

LWWMapTest.kt

same

UnexpectedTypeException.kt

how can this happen? Document it.

crdtlib-kotlin/ src/main/kotlin/crdtlib/*/*

Add asserts on every public method!!!!:

- invariant at entry
- precondition on arguments
- postcondition at exit
- invariant at exit