A Reasonable Plan for RCU Paper

ikuru

July 2018

1 Concrete Essential Refinements

- Making the paper more self-contained
  - Soundness
    - Incorporating Invariants: We have to incorporate the *well-formed memory* invariants to the paper. I think the way we incorporate the memory invariants is quite controversial. As Colin suggested in the meeting, we should first refer to the invariants that are important for the semantics then refer to them when we need during explaning the types/type-rules.
    - Views Constructions: We should bring views constructions back into the paper.
  - Semantics: We should clearly relate our semantics to one of those real implementations. This is crucial.
- Optional-Incorporating the Read-Side-Rules: Depending on how much space we have, we may integrate the read-side type rules and make the type rules complete.
- Why do you need k in loop invariant $\texttt{Next}^k$?: I think this is a very useful feedback! This shows that we should revise our explanation for k. I think consecutive loops in BST-delete example can be a good tool to stress the justification of the usage of k.
- Subtyping: Related with the previous point, I think we should also consider revising the explanation of *sub-typing*.

2 Philosophical Issues

- Simple Assertions: I think we should tone down in our statements for *simplicity of assertions*. This should not mean that we should give it up completely. Instead we should briefly
  - compare the assertions of GPS vs. RCU-Types for a loop traversal

  - mention that the complexity of RCU-types comes due to expressing control-flow.

- Exploring Type Judgements: I think we should also tone up in stating *reasonable type judgements to ensure static guarantees for safe memory deallocation.*

- Client Verification: TONE UP! In aligned with previous two points.