Concurrent Data Structures Linked in Time

Germán Andrés Delbianco, Ilya Sergey, Aleksandar Nanevski, and Anindya Banerjee

Note to Reviewers

This paper has been previously submitted for publication at CONCUR 2016 and ESOP 2017.

Issues with the CONCUR 2016 submission. The submitted manuscript:

`https://arxiv.org/pdf/1604.08080v1.pdf`

The CONCUR 2016 reviewers were concerned about (a) the complexity of the initial specifications, (b) the incomplete mechanization of the proof of snapshot specifications, which at that moment relied on several lemmas, whose proofs were conducted by hand and (c) the lack of client applications.

All of those issues have been addressed since then. To address (a), we simplified the specs significantly, so they would exclude irrelevant implementation-specific constraints on the auxiliary state. For (b), we have finished the mechanized proofs. Finally, to address (c), we added a dedicated section describing client applications of the established specs that are provided in Section 4 of the current submission.

Issues with the ESOP 2017 submission. The submitted manuscript:

`https://arxiv.org/pdf/1604.08080v3.pdf`

No specific technical criticism was raised about this version of the paper, and the reasons for rejection (with the majority of the reviews being positive) were given by the following summary of the PC discussion:

The specifications given in the paper look rather complicated. The introduction dismisses linearizability, but does not give compelling examples showing that the more complicated specifications are necessary.

While we did not change the specifications given in Figure 4 of the present submission from the previous version, we argued throughout the paper that the components of the pre/postconditions from Figure 4 are essential to reason, in a Hoare-style logic, about a large class of interesting concurrent snapshot clients (with examples given in from Section 4)—an aspect that neither of the works on linearizability we are aware of considers. We have added Section 8 with a discussion, comparing our specification and proof method to those of linearizability and arguing for the advantages of our specifications, comparing to seemingly simpler ones. We have also rewritten the introduction, so it would point out shortcomings of linearizability with respect to compositional client-side reasoning, justifying our method that relies on auxiliary state instead. Finally, we extended the related work with a comparison on the concurrent work on proving linearizability using partial orders.

Source Code

Our mechanization of Jayanti's snapshot algorithm and their clients, together with the rest of the FCSL source files and previous case studies is available for download from:

`https://software.imdea.org/~germand/relink`