Automating Proofs of Data-Structure Properties in Imperative Programs

Link: http://arxiv.org/pdf/1407.6124.pdf

Presented Work:

They considered the problem of automated reasoning about dynamically manipulated data structures. The state-of-the-art methods are limited to the unfold-and-match (U+M) paradigm, where predi- cates are transformed via (un)folding operations induced from their definitions before being treated as uninterpreted .They demonstrated the power of our proof rules on commonly used lemmas , thereby close the remaining gaps in existing state-of-the-art systems. We demonstrate the power of our proof rules on commonly used lemmas , thereby close the remaining gaps in existing state-of-the art systems. Another impact, probably more important, is that our method regains the power of compositional reasoning, and shows that the usage of user-provided lemmas is no longer needed for the existing set of benchmarks.

Related Work:

Here is a vast literature on program verification considering data structures. The well known formalism of Separation Logic is often combined with a recursive formulation of data structure properties. Implementations, however, are incomplete, or deal only with fragments. There is also liter- ature on decision procedures for restricted heap logics; we mention just a few examples:. These have, however, severe restrictions on expressivity. None of them can handle the VC's of the kind considered in this paper.

Future Work:

Lemmas can serve many purposes. One important usage of lemmas in U+M systems is to equip a proof system with the power of user-provided re-writing rules, so as to overcome the main limitation of unfold-and-match. However, in the context of program verification, eliminating the usage of lemmas is crucial for improving the performance. This is because lemma applications, coupled with unfolding, often induce very large search space.