Project Report: Verifying Linked List using Z3

Sarat Chandra Varanasi

May 2019

Challenge 1

Choice of Logical system to use: Separation Logic vs My Own Heap Logic. I had to go with the latter as Separation Logic has a huge learning curve. Certain axioms such as *a heap location may be referred only in one fragment of the heap* were counter-intuitive. Although I was able to follow them I was not confident of applying them in the relatively short-span of the semester. Hence I decided to use my own local reasoning. I came up with rules that govern how aliases are formed in the heap, how destructive updates can be performed in a linked list. The idea was to use the properties of the list itself: Reachability, Connectedness and Admissibility of nodes of a linked list. These properties are not assumed globally rather locally. And these notions were sufficient to prove correctness of the Insert and Delete operation.

Challenge 2

Coming up with the notion of admissiblity of a node was easy. But the black-box notion of admissibility was not enough to argue that a *temp* node was inserted into the list. I needed the notions of reachability and connectedness to complete the argument. The reason for this is because a node can be inadmissible in two ways: it is either not reachable or not connected or both. This was difficult to imagine the first time.

Conclusion

This project is a starting point for verifying more data structures. Since my initial focus has been on verifying concurrent data structures, I aim to further explore sequential linked lists with Rely-Guarantee reasoning. Also, the theory must be complete - should disprove all programs that claim to insert, delete an element in a linked list. In practice, the local axioms must be written by a verifier (analogous to tester) with an awareness of what kinds of program might be written by the developer (in terms of the fundamental operations). Main problems are the time it takes to come up with the axioms and the scalability of verification in general for large programs.