Status Report March 15th

jmcamero

March 2021

1 Major Changes

After my last checkpoint meeting it became clear that my project was quickly moving to far into an engineering project instead of what it had intended to be, a research project. A goal over the last two weeks has then been to look into what exactly was the research question that I was hoping to answer and what would be the metrics on which I judge my answer. Since then I have come up with a few different possible questions, each with their own metrics.

2 Dafny Additions

The first question I could consider answering as the aim of the project is based in my work on adding new features to the dafny language that help with writing verifiable code. Specifically I would like to compare adding these features to dafny to adding them to rust, which the group is likely considering transferring to. There are many possible metrics for which to measure this question on, but unfortunately they all seem to lean to heavily on my personal understanding.

3 Understanding Verifiable Code

The next question has to do with challenging a possible misconception that the current research team may be working with which is that verifiable code does not need to be commented as much as may be expected because the verification pre/post conditions act in a way such that the code is ”self documenting”. The question in this case would be to compare some of the functionality found in veribetrfs with a c++ or c implementation and survey others to see if the dafny version of the function is easier to understand. Possible variations of the test may include dafny with and without comments each compared to the other implementation with and without comments.

4 Detecting future verification issues

One of the main issues I have run into is attempting to verify code that already passed verification in the past. This is due to the underlying verification tool z3 involving some randomization based on a seed created by hashing all of the files. What this results in is functions which z3 may be able to verify easily at the moment, but may not be able to verify in the future. My question for this would be is there a possible way to ensure that z3 can prove that a function is verifiable no matter what the input seed is.

5 Next steps

I plan on meeting with my research advisor this week to discuss these questions and possibly others that he thinks may be useful and worthwhile. I should have a formal question by the end of the week that will encapsulate my work that I have already done and the work I am doing for the rest of the semester.