CS 4110 Project Alpha Report

Clay Goodling (jcg284)

November 2020

1 Vision

My current vision for the project is the same as it was when I started. I hope to build a system capable of creating an interpreter for an arbitrary language from a description of its semantics. In light of my first attempt, I have decided not to try to add too many fancy features and instead to try to get it functional at a basic level.

2 Summary of Progress

In this sprint, most of my work was on design and logistics rather than code production. I created some of the infrastructure surrounding building the system, I came up with the syntax for the metalanguage describing a set of semantics, I wrote down the semantics for and a module describing arith, and I began work on the script that will translate a set of semantics into an Ocaml module. The semantics for arith can be found in arith.sem, and the hand-made module describing arith can be found in hand_arith.ml. The beginnings of the script to translate between the two is in module_gen.py. When it is complete, running module_gen.py on arith.sem should result in something very similar to hand_arith.ml.

3 Productivity Analysis

I think I was as productive as I could have hoped to be, but I hugely underestimated the amount of work this project was going to be. Assignment 3 felt fairly mechanical to me, so I figured it wouldn't be terribly difficult to write a program that could do it for me. I realize now that that feeling was likely due to how much of the interpreter was just given to us. I still feel confident that I will be able to get something working by the final sprint, but I am now preparing to commit rather more time to it.

4 Grade

I would give myself a Good grade for this phase. I feel somewhat conflicted about this because on one hand I did not come close to doing everything I said I would do in this sprint in the charter, but on the other hand I put in some 10 hours of work and everything I finished seems to be correct. I think Good is a decent compromise.

5 Goals for Beta

- Satisfactory: Finish the translation from semantics to Ocaml module. Once this is done, all that will be left to have basic functionality will be the parser, lexer, and main modules, and I think that they could be done in the final sprint.
- Good: Satisfactory scope plus the semantics for imp and a number of test programs in arith and imp to make sure the generated interpreters work.
- Excellent: Some amount of work on the parser, lexer, and/or main modules.