COSC561 – M5.8 Programming Assignment 3 - csem

Giuseppe Schintu

University of Tennessee, Knoxville

COSC561 – M5.8 Programming Assignment 3 - csem

With some previous experience on yacc and bottom-up parsing lecture from Module 4, 5, and 6, I started by looking into some documentation for LLVM and its API. I couldn’t figure much after spending about a week. I decided to start working on the csem assignment with a lot of prints and placeholder to figure out what was being called and when. It was intense for the first couple weeks; mostly because the LLVM API does not have examples that would match any of the format in the csem.

I was able to pass input 1,2 and most of 3 in the first 2-3 weeks. However, once I hit input 4, the ADA example in the class for doifelse was not making sense with the backpatch and merge instruction from the “Dragon Book”. I spend another 1-2 weeks going over the m() and n() functions and trying to understand what was going on. I was fortunate enough to get some work on the call and indx (I was able to follow the suggestions from Dr. Jantz announcement).

I am not quite sure when or how it happened, but at some point I started changing a lot of code on the ccexpr and llvm\_op2(my own version to handle the L<->R operations) because my implementation of the CAST (cast\_Rval + 2 overloads) was actually attempting conversions of T\_LBL and T\_ARRAY. I ended up lost for a few days before I stopped and figured out what was really needed from the CAST.

The last 2 weeks, I focused on getting the breaks, continue and the labeldcl to work and use the lscopes[looplevel].breaks and lscopes[looplevel].conts as there is a trace in the variables definitions that I was able to leverage.

It was a stressful experience, but I enjoyed learning about LLVM and I was pretty impressed about the cleanliness and effectiveness that backpatch and merge add to the whole branching process.