

Independent Study: Final Report

Russell Bentley

December 21, 2015

I want this final report to be worth reading. To that end, it seems best to write a short evaluation, and not a long exposition. For this independent study, I set out to write an implementation of Warren's Abstract Machine (WAM), with some goals attached. These goals were, gain experience with prolog implementation, gain experience with C++ and the tools necessary to develop with it, and build a larger piece of software professionally. A reflection on these goals will provide an evaluation of this project and its results.

The greatest success of this project was the exposure it gave me to a variety of tools. A great example is Bison and Flex. These tools were introduced to me in my Compilers course. However, through this project gave I had an opportunity to test my understanding. After a complete rewrite, I was able to build a robust parser for GNU Prolog's WAM code, and build a non-trivial data structure out of it. Having built a parser and a sort of semantic analysis twice now, I feel ready to take on that kind of project again in the future.

The most notable tool though, is C++ itself. In conjunction with work, and my compilers class, I am now far more comfortable with the language. In fact I can now write entire programs, using the standard library, without referring to documentation. C++ has risen to be my goto language for building projects, and I have already used it for several smaller tasks in the last month. This project in particular gave a great deal of experience with low level memory management, which is an important intuition to have when working with systems languages. I was also exposed to `gdb` and `valgrind` for the first time, indispensable tools when working with C++. As a result of this project, I am now a more confident C++ developer.

There are of course other tools worth mentioning. I had to learn about writing Makefiles, which I now use quite frequently. I gained exposure to more of the options available with `g++` (clang++ really.) I used `git` quite frequently, and even my `vim` configuration got some useful tweaks. All told, there was a substantial amount of time spent on this project and much it was used to gain relevant experiences with the tools involved.

The other success was building a familiarity with Prolog implementation. Although it took most of the semester to click, I finally have a good understanding of how the WAM functions. There are two techniques that stand out in particular. The first is the use of choice points to save a previous state during unification. Once I understood this technique, much of the WAM's operation became clear. The second technique of interest is the use of data representation during unification. That is to say, I found the subtlety of building and matching against representations on the heaps intriguing, and it will be the point I continue to think about long after this project. I have also begun to see how one could easily extend the WAM to run external instructions, and support other data types. I set out to learn about prolog implementation and I accomplished that.

Of course for all the positive things I can say, this project had some substantial failures. First and foremost, I did not finish my WAM implementation. In the end I have a program that can match one term predicates. This failure was not alone however. I failed to hit a single checkpoint on time over the semester. I also failed to change my plans to avoid repeating past mistakes. I am still trying to understand where I went wrong. In part, bad time management is to blame. I did not schedule regular hours for this project, which prevented consistent forward progress. I also starting looking at the "big picture" far too late, which rendered much of my early progress useless. All in all, I think this reflects on the goal I set for "professionalism." Estimating the cost of building software is a difficult problem at best, but working effectively is not. Avoiding these failures in future projects will require better planning and time management. Had I been more realistic week to week, there would have been more tangible artifacts to show for my work.

I set out to write an implementation of the WAM. Having not really finished, its fair to say I failed. However, the experience I gained along the way has made the endeavor worth while for me. In addition, I have every intention of getting this project to a more workable state as it is the most substantial piece in my “portfolio” at the moment. I’d say two thirds describes things nicely.