CS 225, Professor Darais
Final Project Proposal
Zachary Bechhoefer, Noah Mintz Roberts

For our project, we plan to write a program that converts C++ into OCaml. Our expectation is that we will define mappings between their syntaxes, and recursively apply those mappings to C++ programs. We believe that the logic of the program will help users translate C++ concepts over to OCaml, and that it is an experiment in technology migration. There is not an expectation that the program will be able to take any C++ program and magically create a replica in OCaml. Rather, we will start by translating the core language, then common header files, and then composing translation-elements together. What we may find is that certain element translate poorly, even though both languages are Turing complete, so our end goal is to translate where possible, and note bad mappings (likely with hardware access) where possible.

Even though their likely isn't a perfect mapping between all C++isms and Ocamlisms, there are workarounds. OCaml has its own C library, allowing hardware access features to be written in C, compiled with *ocamlc* as OCaml modules. In other cases, we may just nest C++isms into classes with override operators (if possible), perhaps implementing ocamlc compiled modules inside them. However, moving beyond one-to-one to one mappings may have side effects on its own. Depending on how special cases turn out, it may be necessary to prove that the OCaml generated by mapping is still Turing complete.