```
To:      Matthias Felleisen, boss
From:    Alex Knauth
         Milo Turner
Subject: Implementing Module / Functor Systems with Macros
```

The Modular Language. We will build a ML-style module and functor system for a simple typed language.

A *module* is either a mod or a functor.

A *mod* is what we normally think of as a "concrete" module, composed of type alias definitions and value definitions. Operations on mods include selecting a value from the mod and sealing the mod with a signature.

A *functor* is a function from module to module. Operations on functors include applying a functor to a module to produce another module and sealing the functor with a signature.

A *signature* as a type for a module.

Motivation

We want to build this prototype as a proof of concept. We plan to move on to adding modules and functors to Hackett if this is successful.

Milestones

- Develop the grammar capabilities of the module language, try to think of edge cases early by creating examples.
- Build the simple typed language.
- Create a module system with just mods and signatures.
- Implement functors and functor signatures.
- Explore integration of this system into the Hackett language.