

Final Tagless While Language, Something Quippy Here

Jeff Young

1 System Overview

This project implements the WHILE Language using the final tagless style. The final tagless style, while briefly discussed in class, is essentially a semantic-based implementation of whatever is being made. In Haskell, this translates into a typeclass-based approach for the WHILE language. For example, instead of having an abstract data type for statements in the language, we will instead have a typeclass with functions that represent statements in our language. In this sense, Final Tagless, allows extensibility in the constructor dimension, because one may extend value constructors and make a new instantiation of the type class, and final tagless allows extensibility in the operator dimension, because one is free to write their own typeclass in final tagless style and instantiate. An important note is that the final tagless style is also extensible in a third dimension, the evaluator dimension, because all of the language constructs are functions in a typeclass, one is free to implement them how they please. Hence, one may choose to implement them with strict evaluation, or lazy evaluation, with a global state or not.

1.1 The Core System

2 Extensibility Scenario

3 Challenges and Design Decisions