Lexically Scoped Hygienic Quasiquotation

Morten Rhiger
Roskilde University, Denmark
`mir@ruc.dk`

September 9, 2012

Abstract

Quasiquotations in Scheme are nearly ideal for implementing programs that manipulate code. They lack only the ability to generate fresh bound variables, as required to make such code-manipulating programs *hygienic*, but any Scheme programmer knows how to address this shortcoming using `gensym`.

In this talk we investigate *hygienic quasiquotations* in Scheme, and in languages influenced by Scheme. Stepping back from implementation issues, we first identify the source of the freshness condition in the semantics of a hygienic quasiquotation facility. We then show how `gensym` is needed to break a meta-circularity in interpreters and compilers for hygienic quasiquotations. Finally, following our recent work, we present a simple an elegant type system for hygienic quasiquotations and we demonstrate that this type system also enforces hygiene using `gensym`.

This talk outlines Scheme programs implementing an interpreter, a compiler, a macro, and a type checker for hygienic quasiquotations.

Keywords Quasiquatations, Program Generation, Hygiene, Lexical Scope, Types

Categories and Subject Descriptors D.3.4 [*Programming Languages*]: Processors—Code generation; D.1.1 [*Programming Techniques*]: Applicative (Functional) Programming