Nikhil Lahoti

012448514

Summary on Sweet.js

The author talks about creating an expressive hygienic macro system for JavaScript like the one for Scheme and Lisp languages. Macro system is not used in JavaScript because of the difficulty of implementing macro system for language without delimited s-expressions. A macro system should be able to manipulate unparsed and unexpanded subexpressions correctly.

They talk about the first step to scheme like macro system is to introduce a read step into the compiler. The way to achieve this is by separating the lexer and parser. This is because, the lexer is not called with the entire running program, but is called in intervals by parser. Parser calls the lexer from a grammatical context to indicate if lexer should consider the / as a regular expression or divide operation. This problem arises because of the shortage of delimiter.

This problem can be solved by building a partial AST during macro expansion. This method then allows macros to manipulate the partial AST. This method is called as enforestation. The authors also added support for infix macros and support for invoke pattern class.

The authors mention the use of declarative and procedural macro definition functionality along with a read algorithm which handles key ambiguities.