**A Summary of “Sweeten Your JavaScript: Hygienic Macros for ES5”Google App Engine REST Server**

Macro systems are used extensively in programming languages such as Lisp and Scheme in large part due to both languages using symbolic (s) expressions to represent expressions and data. These s-expressions make source code easy to manipulate via macros. In contrast, languages with ambiguous grammars are more challenging for macros to refactor; for example, in the case of JavaScript, implementing a reader is difficult due to grammatical ambiguity around regular expressions and the divide operator (“/”). Sweet.js is a macro system for JavaScript whose primary contribution is a reader that correctly distinguish between division operations and regular expressions across the entire ECMAScript5 (ES5) specification; this reader sits between the lexer and parser to eliminate the need for the parser to communicate back and forth with the lexer as defined in the ES5 specification.

While this paper only explicitly describes how to implement the proposed scheme in JavaScript, the approach can be applied to other ambiguous grammar languages. For example, Perl shares ambiguity around the forward slash (“/”) while the Rust language has ambiguity when parsing the less than (“<”) symbol. These types of ambiguities that necessitate intertwining of the lexer and parser can be resolved using the techniques proposed by Disney *et. al.*.