Structuring Languages As Object-Oriented Libraries
P.A. Inostroza Valdera
The development of programming languages is challenging because the typical language processing toolchain consists of different artifacts interacting with each other. The ideal scenario for engineering languages is one closer to traditional software engineering, where reuse, in the form of frameworks and libraries, has optimized development times.

However, in the case of the engineering of programming languages, there is a tension between admitting the evolution of the syntax of languages and at the same time the addition of new processors (or operations). In the literature of programming languages, this tension in terms of extensibility is concisely summarized in the Expression Problem: implement a language for which both new syntactic cases and new processors can be added modularly, without altering the existing code.

Object Algebras solve the Expression Problem in the context of object-oriented programming, requiring only inheritance and parametric polymorphism from the host language. This thesis postulates that, given their advantages in terms of modularity and extensibility, Object Algebras are a suitable technique for structuring libraries of languages from which programming languages can be composed. The ultimate goal is to make reuse a central practice in the implementation of programming languages. This dissertation thus introduces applications of the Object Algebra pattern for implementing language libraries.

First, we introduce Recaf, a framework for creating extensions to Java. In Recaf, a combination of syntactic extensions (supported by a generic transformation) and semantic extensions (provided by an Object Algebra-based embedding) allows us to define libraries of Java dialects.

Motivated by the challenge of improving the usability of transformation-based approaches such as Recaf, we propose String Origins, a technique to trace the results of program transformations back to their origins. Thanks to this technique, for example, the error messages of Recaf-based tools can be greatly improved.

As another application of Object Algebras, inspired by libraries of semantic components, we introduce Modular Interpreters with Implicit Context Propagation. This technique is based on lifting Object Algebra-defined interpreters in order to implicitly propagate context, with the aim of composing modular language interpreters with different context requirements. These modular interpreters can be distributed as libraries of executable semantic specifications.

In Modular Interpreters with Implicit Context Propagation, context is represented using the side-effect facilities of the host language. The lack of control over the interaction between these effects complicates certain scenarios of composition. Motivated by this problem, we introduce JEff, a language that integrates object orientation with effects and handlers, as a first step towards an object-oriented language that could host effectful modular interpreters.

The results described in this dissertation show that object orientation, and Object Algebras in particular, provides a fertile foundation for the vision of library-based language development.