A System for the Generation of objectoriented Compilers

The thesis presents the *OCC* system (*OCC* stands for Objectoriented Compiler Construction) which supports the generation of compilers for imperative and objectoriented programming languages. The support is concentrated on important tasks during the definition and development of new languages e.g. fast implementation of compiler prototypes through reuse of existing compiler components for specific language concepts, exchange of components and integration of new concepts into the existing ones.

At first important imperative and objectoriented programming languages are checked for common properties. A language is considered important if it is (or was) widely used or has introduced new concepts that influenced the evolution of other programming languages. The common concepts of these languages were integrated into a language independent model. On this model a library of classes is based that implement language concepts frequently needed in compilers.

The *OCC* system is centered around the mentioned class library. In order to implement a specific programming language the developer of the that language establishes some relations between the constructs of the language and the classes of the library. A tool supports the adaptation of a language construct to one or more library classes. After the adjustment methods of the library classes perform the semantic analysis and the code generation for the concepts of the new language. The use of multi inheritance allows to parametrize the semantic analysis of language constructs according to the specification of the language; specializations of existing concepts may be implemented through inheritance in an adequate way. It is possible to exchange different language concepts within distinct categories without affecting the rest of the compiler. This is achieved by modeling the library classes as different implementations of abstract data types. The system is completed by a hypertext documentation that supports the developer to find language concepts and to reuse them.

The *OCC* system is compared to several other systems and environments that support the generation of compilers or at least of some compiler phases.

Selected aspects of a prototype *OCC* implementation are presented and explained. The Implementations of two different languages (Oberon-2 and W-Lisp 3) demonstrate the practicability of the prototype.