# Abstract

Function inlining has been considered an important optimization method for compiled languages. In this thesis, we especially study the inlining benefits and present an object-oriented implementation of a source-level C language inliner. Various parsing and inlining topics, such as: message driven system, design patterns, parse-tree and symbol-table organizations, memory management strategy, program standardization and function-inlining steps are carefully presented.

For efficiency and ease of implementation, many optimizing compilers implicitly impose an "inlining-decision algorithm" to restrict the conditions under which a function invocation could be inlined. Source-level inlining, profile-guided inlining, cache issues and the version issue have lead to various conflicting decision algorithms. To overcome previous inlining work, an ambitious inlining-decision algorithm combined these techniques, which creates an automatic C inliner.