
発表概要

Calc コンパイラの作成

浜田 陽子[†] 渡辺 理沙[†]
坂下 徳子[†] 浅井 健一[†]

簡単な関数型言語 Calc を設計し、そのコンパイラを作成したのでその模様を報告する。Calc は 1 階の関数型言語で数値計算を行うことを念頭に、代入可能な配列を扱うことができる。Calc のプログラムは Calc コンパイラによって字句解析、構文解析を経て、k-正規形と呼ばれる中間言語に変換される。最適化には変数名のつけかえ、不要変数除去、定数伝搬などの通常のものに加え、このコンパイラ特有の最適化として部分評価に匹敵する強力な関数展開を備えている。さらに、コード生成部においては、末尾呼び出し最適化や配列アクセスの埋め込みを行っている。コンパイラの構造は従来の最適化に比べて比較的簡明であるが、実際にこの上でキャビティフローを計算する数値シミュレーションプログラムを動かしたところ、C に近い性能で実行することができた。

Design and Implementation of Calc Compiler

YOKO HAMADA,[†] RISA WATANABE,[†] NORIKO SAKASHITA[†]
and KENICHI ASAI[†]

We report on the design and implementation of a compiler for a simple first-order functional language Calc. Calc is equipped with assignable arrays with computing numerical analysis in mind. After going through lexical analysis and syntax analysis, programs are first transformed into an intermediate language called k-normal forms. The Calc compiler then performs ordinary optimizations such as alpha-transformation, elimination of useless variables, and constant propagation. Furthermore, it performs strong inlining (or partial evaluation) as an optimization particular to Calc. In code-generation phase, it performs the tail-call optimization as well as embedding of array access primitives. Although the structure of this compiler is rather simple compared to ordinary compilers, it produces fairly efficient code. In fact, we can execute a numerical simulation program calculating cavity flow with efficiency comparable to the same program written in C.

(平成 15 年 6 月 16 日発表)

[†] お茶の水女子大学
Ochanomizu University