
発表概要

連結生存区間に基づくレジスタ割付け方式の提案

渡 邊 坦†

レジスタ割付けは、従来、ソースプログラムの変数やコンパイラの生成する一時変数にレジスタを割り付ける問題としてとらえられ、その方法としては、任意の2つの変数に対して、その生存区間が重なり合うならば互いに異なるレジスタを割り当てるグラフ彩色法が著名である。レジスタは主として変数や部分式の値の入れ場所として使われ、コード最適化を行わない場合、レジスタの生存区間はきわめて短く、レジスタが32個以上あるような計算機では、レジスタ溢れほとんど発生しない。本発表では、レジスタ溢れはコード最適化にともなって発生すると見て、レジスタ割付けを最適化と連動させ、溢れコストを低く抑えながらレジスタの生存区間を延長させてゆくレジスタ割付け方式を提案する。具体的には、1つの同型部分式に1つの抽象レジスタを対応させる単一表現レジスタと、大域的共通式削除にともなって成長する連結生存区間という概念を導入し、各定義点ごとに抽象レジスタの別名を発生させ、共通式削除にともなって別名レジスタを合体させながらレジスタ生存区間を連結してゆき、コスト計算に基づいてその生存区間を区切る形をとるレジスタ割付け方式を提案する。この連結生存区間ごとに実レジスタを割り当てることにより、溢れコストを低く抑えることができる。

Proposal of a Register Allocation Method Based on Connected Live Range

TAN WATANABE†

Register allocation has been treated as a problem of allocating registers to source program variables and compiler-generated temporals. A well-known method is graph-coloring algorithm which assigns registers so that any 2 variables have different register if their live range overlap with each other. On RISC machines, registers are mainly used to hold loaded values and computed results. This means that register live ranges are very short if code optimization are not performed and register spilling will rarely happen for computers with 32 or more registers. Seeing that register spilling will be caused by code optimization, this paper presents a register allocation method that extends register live range along with code optimization suppressing the cost of spilling registers. In more detail, a notion of single expression register assigned for each subexpression having the same form and a notion of connected live range that grows according to global common subexpression elimination are introduced. At each definition point of the abstract register, an alias name is generated and they are coalesced along with common subexpression elimination forming connected live range and the live range is terminated at a point of low spilling cost. By allocating real register for each connected live range, spilling cost of registers can be suppressed.

(平成12年11月16日発表)

† 電気通信大学情報工学科

Department of Computer Science, The University of
Electro-Communications