

等式分割を用いた配列データ依存解析手法

4 X - 2

小山洋一[†], 國枝義敏^{††}, 湯淺太一[†]

[†]京都大学大学院工学研究科情報工学専攻, ^{††}和歌山大学システム工学部情報通信システム学科

1 はじめに

逐次的な手続き型言語のプログラムを自動並列化するには(多重)ループ中に含まれる配列操作に対してデータ依存解析を行い並列実行可能かどうかを判定する。つまり参照の添字が繰返し空間内で同じ値を取り得るかを判定する必要がある。一般には不等式の制約下で連立方程式を解く問題となる。このデータ依存解析の手法としてはGCDテスト, Banerjeeの不等式の手法[1], Omegaテスト[2], V-Pascal[3]の手法などがある。このうち後二者は上下限式や添字式が制御変数の線形式である場合に厳密な解析が可能だが, これらの手法は解析にかかるコストが問題であった。そこで本研究では, それぞれの変数の上限と下限の制約が静的に決定できる時に一つの整等式が複数の整等式からなる連立方程式に分解できる場合があることに着目して, 配列データ依存解析の効率化手法を提案する。

2 等式分割

配列データ依存解析の入力としてPascalのfor文のような多重ループを考える。各制御変数は上下限の制約を持っており, 整数値のみを取る。上下限の制約式および配列参照の添字式がともに整数係数からなる線形式である場合, 添字式同士を結んだ等式中の各項のとりうる数値の範囲には規則的な偏りができる。これを利用して, ある等式を, 元の等式の解と同じ解を持つような複数の等式に分割することができる場合がある。これを等式分割と呼ぶ。等式分割により連立方程式中の各等式を複数の等式に分割できれば, 独立の等式の数が増え, その結果連立

方程式全体の解の自由度が下がる。これにより早い段階で依存の有無を判定することが期待できる。

例えば, 整式 $f(i_1, i_2, \dots, i_n) = 0$ が, 整式 f_1, f_2 を用いて以下のように表せる場合:

$$\begin{cases} N f_1(i_1, i_2, \dots, i_n) + f_2(i_1, i_2, \dots, i_n) = 0 \\ -N + 1 \leq f_2(i_1, i_2, \dots, i_n) \leq N - 1 \end{cases} \quad (N \text{は} 2 \text{以上の自然数})$$

$f = 0$ を $f_1 = 0, f_2 = 0$ の二つの等式に分割して考えても, 同じ解が得られる。元の等式の係数の絶対値に大きな偏りがある場合にこのような N を見つけることができる。上記のように係数の絶対値に大小の偏りがある場合だけでなく, 整式の係数の絶対値が二種類あってそれらが互いに素な場合も同様にして等式分割を考えることができる。

また, コンパイル時には値が決定できないような変数(記号定数と呼ぶ)を係数に持つ場合にも, その記号定数を含む項と含まない項に等式分割できる場合がある。これはFFT(高速フーリエ変換)のプログラムなど多くのプログラムにおいて頻出するパターンだが, 記号定数を含むが故に既存の手法の多くは静的に解析できなかった。本研究の手法を用いることにより, この種のパターンにおいても依存の有無を静的に解析できる。

参考文献

- [1] Banerjee, U.: Dependence Analysis for Supercomputing, *Kluwer Academic* (1988).
- [2] Pugh, W.: The Omega Test: a fast and practical integer programming algorithm for dependence analysis, *Proceedings Supercomputing*, pp.4-13 (1991).
- [3] Tsuda, T., Kunieda, Y. and Uehara, T.: The Vectorizing/Parallelizing Compiler V-Pascal, *Fifth Workshop on Compilers for Parallel Computers*, pp.128-137 (1995).

A Method for Array Dependence Analysis Using Function Decomposition

Youichi Koyama[†], Yoshitoshi Kunieda^{††}, Taiichi Yuasa[†]

[†]Kyoto University, ^{††}Wakayama University