レジスタ生存グラフを用いたレジスタ割付け手法の改善

4U－6

近藤 伸宏† 古関 聰‡ 小松 秀昭‡ 深澤 良彰†
†早稲田大学理工学部 ‡日本IBM(株) 東京基礎研究所

1 はじめに

一般的に用いられているレジスタ割付け手法としては、グラフ彩色法を用いたものがある。これに対し、レジスタ生存グラフを用いた手法が提案されている[1]。

本稿では、レジスタ生存グラフをシリーズパラレル型のグラフに変形し、グラフ内のノードのグループ分けを容易にすると共に、重要度の高い部分の見積りを行ない、それに基づいてレジスタ割付けを行なう手法を提案する。これにより、レジスタ割付けの速度を落さずに、従来手法よりも性質のよいレジスタ割付けが可能となる。

2 本研究の概要

文献[1]では、レジスタ生存グラフの有用性が示されている。そこで用いられているレジスタ割付け手法では、ノードの自由度が小さいものを優先してレジスタに割り付けている。しかし、スピルコードを挿入する位置の決定法が、自由度でソートされたリストのうち割付けができないものになっているため、必ずしも適切な位置にスピルコードが挿入されない場合があった。本研究では、レジスタ割付けの発見的手法の改善によってこの欠点を補う。

本手法では、レジスタ生存グラフに対してノードのコピーなどを用いて変形を施し、シリーズ接続（直列接続）とパラレル接続（並列接続）を再帰的に適用して構成される、シリーズパラレル型のグラフ[2]を用いる。これによって、レジスタ生存グラフ内のノードのグループ分けとそのグループに対する意味づけが行なわれ、それに基づいた優先度を用いてレジスタ割付けを行なうことが可能となる。

3 アルゴリズム

3.1 準備

まず、用語を定義する。図1は、レジスタ生存グラフ上のノードをSグループおよびPグループでまとめた時の図の一部である。レジスタ生存グラフは最内ループに含まれる各シンボリックレジスタをノードとし、シンボリックレジスタの値の使用に関するエッジを張ったグラフである。

Sグループ シリーズパラレル型のグラフに簡素化されたレジスタ生存グラフにおいて、連続したシリーズ接続の部分をひとまとめにしたもの。

Pグループ シリーズパラレル型のグラフに簡素化されたレジスタ生存グラフにおいて、連続したパラレル接続の部分をひとまとめにしたもの。

高さ シリーズパラレル型のグラフに簡素化されたレジスタ生存グラフにおいて、Sグループの高さとは、そのSグループが属しているPグループのネストの数。

自由度 Pグループ毎に定義され、そのPグループに含まれる全てのSグループそれぞれについてそのSグループ内の命令を全て実行するのにかかる最短時間を求め、その値が最も大きいSグループの自由度を0とする。他のSグループの自由度はそのSグループ内の命令の最短実行時間と、自由度0のSグループ内の命令の最短実行時間との差とする。

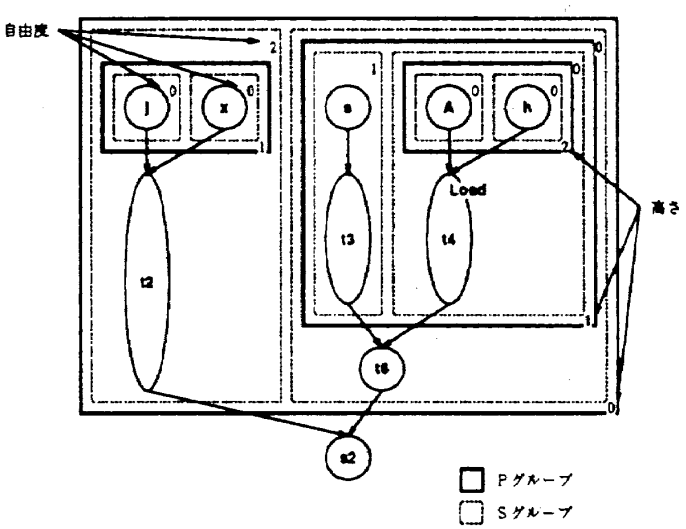

図 1: 用語の定義

Improvement of Register Allocation Technique using Register Existence Graph
Nobuhiro Kondo†, Akira Koseki‡, Hideaki Komatsu‡, and Yoshiaki Fukazawa†
†School of Science & Engineering, Waseda University
‡IBM Japan.Ltd. Tokyo Research Laboratory

3.2 アルゴリズム

本レジスタ割付けアルゴリズムは、シリーズパラレル型のグラフに簡素化されたレジスタ生存グラフ上で行なわれる。

最初に、対象となるグラフをSグループ、Pグループでグループ化する。シリーズパラレル型のグラフは、グラフ上のある経路がシリーズ接続であるか、パラレル接続であるかということをノードにした2分木で表現できることが知られている。このことと、SグループおよびPグループの定義から、複数のSグループはPグループによってまとめられ、複数のPグループはSグループによってまとめられる。

ここで、Sグループ、Pグループの性質について考えてみると次のことが分かる。

Sグループとは、ある一連のタスクで使用する変数の集まりである。これは、あるSグループに注目した時に、その中で使用されている変数のうち、Sグループの入口と出口にあたる部分の変数以外は、そのSグループ内のみでしか使われないからである。したがって、同じSグループ内の変数は同時にレジスタ割付けの対象とした方が、そのタスクが最大限円滑に実行できるような割付けが可能になるものと考えられる。

Pグループは、ある時間に同時に計算がれるべきSグループの組を示す。これは、あるSグループの出口で使われているレジスタの値が、そのSグループの属するPグループの終点である接続点において、同じPグループ内の他のSグループの出口で使われているレジスタの値と共に活用されているからである。したがって、同じPグループ内の複数のSグループのタスクで重要なことは、全てのSグループのタスクが終了した時間であり、必ずしも全てのタスクが最速で終っている必要はない。

以上の性質を考慮した、レジスタ割付けのアルゴリズムの概要は次のようになる。

1. シリーズパラレル型に簡素化されたレジスタ生存グラフを、Sグループ、Pグループでグループ化する。
2. 全てのSグループのリストを作り、高さが高い順にソートする。
3. 同じ高さのSグループが複数ある場合、自由度を求め、その自由度の低い順にソートする。
4. 同じ高さ、同じ自由度の場合、同じPグループに属する他のSグループのうち、レジスタ割付けが完了しているグループの数の多い順にソートする。
5. Sグループのリストの先頭要素のレジスタ割付けを行なう。
6. 5でレジスタ割付けが完了したことにより、あるPグループ内の全てのSグループのレジスタ割付けが完了した場合、そのPグループの終点である接続点のレジスタを割り付ける。
7. 3～6をリストから要素がなくなるまで繰り返す。

また、ここで用いられるレジスタ割付けの規則を以下に示す。規則中で時間が用いられているが、これは最初に割り付けられたレジスタの位置を基準にして、レジスタ生存グラフの時間軸と、レジスタ割付け後のグラフの時間軸を重ね合わせて定められるものである。

- レジスタの干渉度が実レジスタ数を越えない限り、レジスタ生存グラフ上の位置に対応する時間に割り付ける。
- レジスタの干渉度が実レジスタ数を越えた場合、スピルコードを挿入する。スピルインの位置は、次のルールに基づき決定する
  - 同じPグループ内の他のSグループのレジスタ割付けが終っていない場合、可能な限り早い位置に挿入する。
  - 同じPグループ内の他のSグループのレジスタ割付けが終っている場合、そのSグループの実行の終了位置に合わせ、可能な限り遅い位置に挿入する。
- シリーズパラレル型のグラフに簡素化する際に分割された変数が、他のSグループにおいて利用されるために既にレジスタに割り付けられている場合には、それを利用するか、利用できない場合はスピルコードを挿入する。

4 おわりに

本稿では、レジスタ生存グラフを用いたレジスタ割付け手法の改善について述べた。本手法では、まずレジスタ生存グラフをシリーズパラレル型のグラフに簡素化し、ノードをグループ分けする。そして、そのグループ毎の優先度に基づいてレジスタ割付けを行なうことによって、コンパイル速度を落さずに性質の良いコードを得ることができる。

本稿では高さによって重要度を決定したが、実際には高さが増す時にどのくらいの影響が出るのかということを指標にするのが望ましいと思われる。今後は、高さが増した時のコスト計算をDPを用いて見積もることによって、アルゴリズムの改良を行なっていきたい。

参考文献

[1] 古関聰, 小松秀昭, 百瀬浩之, 深澤良彰：" 命令レベル並列アーキテクチャのためのコードスケジューラ及びレジスタアロケータの協調技法", 情報処理学会論文誌,Vol.38,No.3,1997, pp.584-594.

[2] R.J.Duffin:"Topology of Series-Parallel Networks", J. Math. Applic. 10, 1965, pp303-318