

手続き間解析を用いたグローバル変数の フローセンシティブな依存の解析方式

1 E - 4

新名 博 田村智洋 藤原優史 岩澤京子
東京農工大学 工学部 電子情報工学科

1. はじめに

手続き間の情報を含めたデータフロー解析を行うことで、手続き呼出しを含めた最適化変換を行うことができるようになる。本研究では、自動並列化変換で利用できるような、手続き間の大域変数の依存関係を生出力するための解析を行うことを目標とした。

2. 機能

構文規則を次のように限定した C 言語を入力として中間語を作成し解析を行う。

- ・制御構造として、if 文、for 文、while 文、do-while 文が使用できる。
- ・一次元配列を使用できる。
- ・値を返さず、引数を持たない手続き呼出しができる。

さらに、データフロー解析では次のような制限を行った。

- ・変数のみを解析する。
- ・再帰呼出しを対象外とする

上記の条件を満たすソースプログラムに対して、中間語を作成し、到達定義と露出使用⁽¹⁾を解析する。また、問い合わせ関数によって、到達定義と露出使用から連鎖を生出力する。構成は、データフロー解析部と問い合わせ関数に分けることができる。自動並列化変換システム上での位置づけは、図 1 のようになる。

3. 解析の方法

各手続き呼び出しでは、呼び出し先の手続きのデータフロー情報を取り込む必要があるため、コ

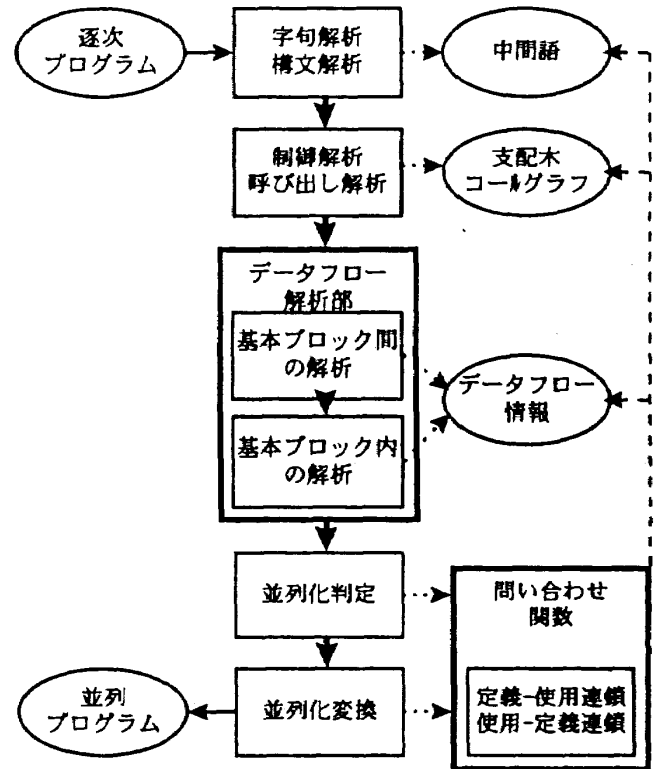


図1 システムの構成

ールグラフの末端にあたる手続きから順に解析を行う。解析は、呼び出される手続きの解析、呼出し元での解析、定義-使用連鎖・使用-定義連鎖の作成の三つに分けられる。

(1) 呼出される手続きの解析

- ・手続きの入り口の点ではすべての大域変数の定義が到達しているとする。
- ・手続きの出口ではすべての大域変数の使用が露出しているとする。
- ・基本ブロックの入り口の到達定義は、直前の基本ブロック全ての出口の到達定義とし、入り口の到達定義から基本ブロック内で消滅する定義を除き、生成される定義を加えたものを出口の到達定義にする。

・基本ブロックの出口の露出使用を、直後の基本ブロック全ての入り口の露出使用とする。出口の露出使用から基本ブロック内で消滅する露出使用を除き、生成される露出使用を加えたものを入り口の露出使用とする。

(2)呼出し元での解析

- ・call文だけの基本ブロックを作成する。
- ・呼び出される手続きの出口の到達定義を取り込み、基本ブロックの入り口の到達定義から、呼び出し先で消滅する到達定義を除き、生成される到達定義を加えて、基本ブロックの出口の到達定義にする。
- ・呼び出される手続きの入り口の露出使用を取り込み、基本ブロックの出口の露出使用から、呼び出し先で消滅する露出使用を除き、生成される露出使用を加えて、基本ブロックの入り口の露出使用にする。

この解析により、図2のプログラムを入力とすると、データフロー解析の結果は図3のようになる。

(3)使用-定義連鎖と定義-使用連鎖

問い合わせのあった変数の定義や使用に対して、露出使用に含まれる使用や到達定義に含まれる定義をリストにする。手続き呼び出しの先の定義や使用も含めたものをリストにできる。

図3から、図4のような連鎖が作られる。

4. 評価

本研究の解析方法では、呼び出しの点で呼び出し先のデータフロー情報が反映されるように、情報の生成を行っている。解析では呼び出す手続きをインライン展開したものと同様の結果を得られる。

5. おわりに

手続き間解析を用いることで、ループ内に手続き呼び出しを含んでいても、並列化が可能になる。今後は、手続き呼び出しに引数持たせることなどの拡張が必要である。

```

int a, b, c;
main(){
  c = 2;
  call p();
  c = b;
}

p(){
  a = 3;
  call q();
}

q(){
  b = c + a;
}
    
```

図2 機能を限定したC言語のソースプログラム

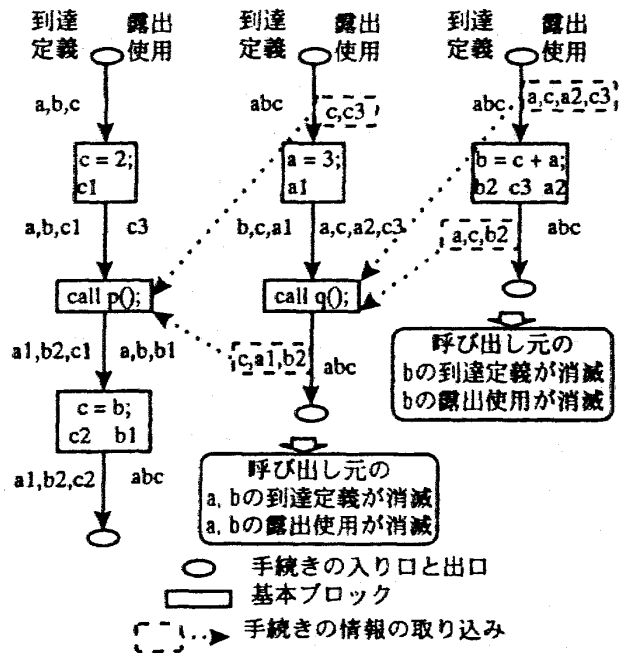


図3 データフロー解析の結果

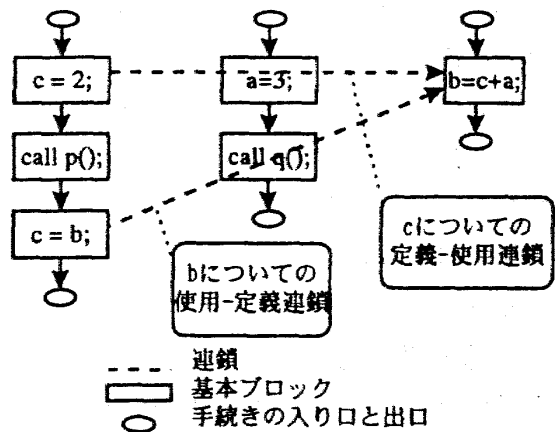


図4 連鎖の例

参考文献:

[1] A.V.エイオ,R.セシイ,J.D.ウルマン,原田賢一(訳): コンパイラII -原理・技法・ツール-, サイエンス社, 1990