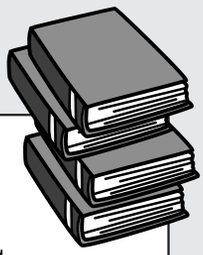

R. Cytron and J. Ferrante and B. K. Rosen and M. N. Wegman and F. K. Zadeck：Efficiently Computing Static Single Assignment Form and the Control Dependence Graph

ACM Trans. on Programming Languages and Systems, Vol.13, No.4, pp.451-490（Oct. 1991）

本論文は，最適化コンパイラにおいて利用されるようになってきたSSA形式（Static Single Assignment Form: 静的単一代入形式）への効率的な変換のアルゴリズムについて述べたものである☆1．SSA形式とはコンパイラにおけるプログラムの表現形式の1つで，プログラム中のすべての変数の使用（use）に関して，その利用の定義（def）が1つしかないように表現したもので，これにより変数の値の定義と使用の関係が明確になり，それまでの個々の変数の定義と利用を表現したdef-use chainを用いたデータフロー解析とは異なる方法でのさまざまなコード最適化のアルゴリズムを考えることができる．

プログラムをSSA形式に変換するためには，代入ごとに変数の名前をつけなおしていけばよい．たとえば，$a=\ldots$；　$a=a+1$;　$b=a+4$; というコード列に対しては，$a1=\ldots$；$a2=a1+1$;　$b1=a2+4$; というように名前を変える．基本ブロック内ではこの操作は明らかであるが，分岐や合流がある場合にはϕ関数という仮想の演算子を用いる☆2．制御フローグラフにおいて，合流する2方向から変数aに対する定義が達する場合には，それぞれの方向から到達する定義で変えた変数名$a1$，$a2$に対し，合流の先頭において，$a3=\phi(a1, a2)$を挿入し，以降$a3$を変数aに対する名前とする．

SSA形式の利点の1つは変数の定義・利用に関する情報が集約されており，直感的に分かりやすいという点である．この発想のもとになっているアイディアは，基本ブロック内において計算された値に対し番号付けを行うvalue numberingからきたもので，ϕ関数を導入することにより，これを基本ブロックを超えて手続き全体の解析に容易に拡張できる（global value numberingと呼ばれている）．SSA形式が初めて提案されたのは1988年のPOPL（Annual Symposium on Principles of Programming Languages）で発表された，RosenらのSSA形式を使った冗長コードの削除の論文[1]と，AlpernらのSSA形式で条件分岐にまたがる共通部分式の削除の論文[2]の2件の論文で同時に提案されている．両論文とも，非常に明解で読みやすい論文なので，SSA形式に興味がある方は参照いただきたい．これらの論文で示されているように，SSA形式により基本ブロック内で適用していたアルゴリズムを容易に手続き内の大域的な最適化に拡張が容易かつ効率的になる．

本論文は，SSA形式への変換をdominance frontiers☆3という制御フローグラフ上の性質を使って，効率的に求める方法を提案したもので，1989年のPOPLで初めて発表された内容[3]を含んだものである．同時にプログラムの大域的な最適化に重要な概念である制御依存グラフ（Control dependency graph）についても述べている．

コンパイラを実際に開発する場合，まず考えなくてはならないのがコードに関する情報を表現するためのデータ構造であり，SSA形式は非常に有効なデータ構造となる．コンパイラの教科書というと，古典のAho&Ullmanのいわゆる Dragon Bookがあるが，本を読んだだけではなかなか実際のコードが書けない．筆者が数十年前にコンパイラを開発する際に出会ったのがRosenらの論文[1]で，SSA形式のプログラム表現による最適化の明解さと能力に感銘を受けた．SSA形式は，グラフ彩色による大域レジスタ割り当てと並んで，Dragon book以降のコンパイラ技術の大きな進歩の1つといえると思う．

コンパイラは計算機科学の中でも最も古い分野の1つであるが，高性能プロセッサから組み込み用プロセッサまで計算機アーキテクチャの進歩が続く限り，重要かつ基礎的な分野であることは論を待たない．新たなコンパイラを開発することは容易ではないが，これからのコンパイラにはSSA形式などの技術が多く取り入れられていくであろう．

なお，SSA形式が述べられている日本語の文献としては中田先生の著書[4]があることを付け加えておく．

参考文献

1）Rosen, B. K., Wegman, M. N. and Zadeck, F. K.: Global Value Numbers and Redundant Computations, Proc. of 15th POPL, pp.12-27（1988）.
2）Alpern, B., Wegman, M. N. and Zadeck, F. K.: Detecting Equality of Values in Programs, Proc. of 15th POPL, pp.1-12（1998）.
3）Cytron, R., Ferrante, J., Rosen, B. K., Wegman, M. N. and Zadeck, F. K.: An Efficient Method of Computing Static Single Assignment Form, Proc. of 16th POPL, pp.25-35（1989）.
4）中田育男，コンパイラの構成と最適化，朝倉書店（1999）.

☆1 筆者は実際のベンダのコンパイラがどの程度SSA形式を利用しているかは知らないが，いくつかのコンパイラでは実際に利用されているようである．

☆2 ϕ関数はデータフロー計算モデルのマージ演算子に似ているが，あくまでもコンパイラの中間表現上のものであり，異なるものである．

☆3 制御フローグラフにおいて，ノードXのdominance frontiers（支配辺境）とは，ノードXの支配（dominate）関係にあるノード集合に隣接するノード（つまり，支配関係でなくなるノード）の集合のことである．

（平成16年8月18日受付）

佐藤三久／筑波大学
msato@is.tsukuba.ac.jp