

Title	関数型プログラムにおけるプログラム変換の研究
Author(s)	村島, 康哲
Citation	
Issue Date	1999-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1257
Rights	
Description	Supervisor:外山 芳人, 情報科学研究科, 修士

関数型プログラムにおけるプログラム変換の研究

村島 康哲

北陸先端科学技術大学院大学 情報科学研究科

1999年2月15日

キーワード： 関数型プログラム, プログラム変換, 切り払い.

プログラム変換とは、「あるプログラムから、その正しさを保持しつつ、効率化された新たなプログラムを得る」という事を目的としている。プログラム変換において、変換の過程でプログラムの意味が変わらず、計算速度も改善されていること、すなわち広い意味での正当性(以下、正当性と略す)が保証されることが重要である。このような変換は、効率的なプログラムの開発だけでなく、変換を通して新しいアルゴリズムの開発に寄与することができる。

関数型プログラミングは、基本的な関数を定義し、それらの合成によってプログラムを作成する。基本的な関数はリストや木といったデータ構造上で定義されることが多い。関数の合成により、数多くの中間データが作成される。だが、このようなデータ構造を計算途中で生じさせるプログラムは、それらを生じさせないものと比較して一般に計算効率が悪い。そこで、不必要な中間データを除去することを目的として Wadler(1990) は、切り払い(deforestation)と呼ばれるプログラム変換法を提案した。

従来の切り払いの研究の多くは、遅延評価(lazy evaluation)に基づく言語上でなされてきた。一方、先行評価(eager evaluation)に基づく言語上では、あまり研究されていない。切り払いが先行評価に基づく言語を対象に行なわれないのは、変換規則の中に、正当性が保証されない規則がある事が原因である。

先行評価に基づく言語を対象にした切り払いとして、佐賀(1998)は部分計算的切り払いを提案した。部分計算的切り払いでは、変換の停止性と、変換前後のプログラムの等価性(正当性からステップ数の改善に関する性質を除いた性質)が保証されている。部分計算的切り払いは、切り払いが先行評価言語においても有効であることを示している。しかし、ステップ数の改善に関する理論的な考察はなされていない。

また、言語の評価機構にかかわらず、Wadler や佐賀の結果を含む多くの研究成果では、関数定義の右辺は線形な項に制限されている。このことは、プログラムを作成する上で非常に大きな制限になっている。

本研究では、先行評価言語において、正当性を保証する新しい変換方法を提案する。本研究では、Wadler の提案した印付き切り払いをもとに、先行評価言語に適した切り払いを提案する。

印付き切り払いは従来の切り払いと比較して次のような特徴を持つ。

- 中間データとして不都合が起こらない項（数値型、ブール型）は中間データとして現れても良い。そして、それらの項は関数定義の右辺において線形でなくても良い。よって、より幅広いプログラムを作成することが可能である。

さらに、部分計算的切り払いと比較して次のような特徴を持っている。

- 部分計算的切り払いの対象となる言語では、足し算やかけ算などの原始関数も再帰的に定義する必要があった。一方、印付き切り払いの対象となる言語では、原始関数をあらかじめ用意された組込関数として取り扱うことができる。よって、印付き切り払いは、より現実的なプログラムに対する切り払いを考えることが可能であるといえる。

しかし、印付き切り払いを、先行評価プログラムに対してそのまま用いると、変換の正当性が成り立たなくなる。さらに、リストや木などの型を持つ変数は項の中で線形でなければいけない。

これらの問題点を解決するために、まず先行評価言語で印付き切り払いの変換規則を適用したときに正当性が保証されない規則を特定した。そして、印付き切り払いの 11 個の規則のうち、関数の展開と代入に関する規則を適用する場合に正当性が保証されなくなることが分かった。そのような場合、次の 2 つの操作によって正当性を保証できることを示した。

- 変換規則を適用するために満たすべき条件を付加する。遅延評価言語を対象とした場合、無条件で適用しても正当性に影響はでなかった。しかし、先行評価言語を対象とした場合、条件づけを行なう必要がある。
- 関数の展開に関して、変換終了後に新関数を導入するという操作を行なうことによって展開に関する正当性を保証することができる。

線形の制限を外すことに関しては、印付き切り払いにおける線形でない項（数値型、ブール型）の切り払いの操作を参考に変換の方法を考案した。展開の規則を適用する時に、関数定義の右辺において線形でない変数の位置にある項は、Let 式を用いて先に評価した方が効率が良い。さらに、先行評価の戦略の特徴を利用して、関数を展開する時に複製される項以外でも共通する計算を 1 回でまとめるような変換を提案した。切り払いは、評価の途中で生成される中間データを取り除く変換方式だが、先行評価においては逆に中間データを作ることによって変換の正当性が成り立つ場合もあるということが分かった。ただし、関数を展開する時に複製される項以外の共通項を取り出すという操作は正当性が保証される範囲の中で行なわれる。

本研究では、先行評価向け印付き切り払い手続きの停止性、正当性を示した。そして、実際に先行評価向け印付き切り払いで変換を行ない、正当性が保証されていることを確かめた。ただし、変換の効率は入力されるデータの大小に左右されるため、一般的な効率の向上率といった指標を得ることはできなかった。

以上、本研究では、先行評価言語向け印付き切り払いという新しい変換手法を提案し、その有効性と問題点、その解決策について検討した。本研究で検討を行なったことから、先行評価言語における切り払いを高階関数に対する切り払いなど拡張する時にも有効であると思われる。