

Title	型理論に基づくプログラミング言語の効率的な実装に関する研究
Author(s)	挽地, 篤志
Citation	
Issue Date	2002-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1533
Rights	
Description	大堀淳, 情報科学研究科, 修士

型理論に基づくプログラミング言語の 効率的な実装に関する研究

挽地篤志 (010092)

北陸先端科学技術大学院大学 情報科学研究科

2002 年 2 月 15 日

キーワード: type directed compiler, A-normal form, Logical Abstract Machine.

1 導入

従来、コンパイラの開発は ad hoc になされてきた。そのため、高級言語から低級言語にコンパイルできたとしても、実行時に誤りを含む可能性があった。それに対して近年、論理学とプログラミング言語の関係が分かってきた。論理学とプログラミング言語における命題と型、証明とプログラムの間には関係がある。論理学において P が命題 A の証明であるなら、プログラミング言語において P は型 A の値を計算するプログラムであると言える。本稿ではこの論理学とプログラミング言語の関係をを用い、コンパイラの分析と実装を行う。以下、既存の研究とその問題点、本研究の目的、本稿の構成について述べる。

2 既存の研究と問題点

コンパイラとは、ソース言語から中間言語を経て目的言語へと変換するプログラムのことを言う。

コンパイラの研究には、中間言語に CPS を使ったコンパイラ [Appel et al.: 89]、A-normal form を使ったコンパイラ [Flanagan et al.: 93] がある。Appel は、中間言語に CPS を使ったコンパイラを設計した。CPS は継続 (Continuation) と環境 (Closure) を引数にとる中間言語である。CPS の欠点は、継続を引数に取ることによってコード量が増大することである。それに対し Flanagan は、中間言語に A-normal form を使ったコンパイラを設計した。A-normal form は中間の値に名前を付けるという特徴を持つ中間言語である。A-normal form の利点は、コード量が増大しないということである。これらの研究により、A-normal form は CPS よりも良い変換を与えることが分かった。しかしこれらのコンパイラは型無しの言語を対象にしているため、論理学との関係を適用することができない。

それに対して最近、型付きのコンパイラの研究がなされてきた。型付きのコンパイラの研究には、型主導のコンパイルの研究 [Morrisett et al.: 98] がある。Morrisett の研究は、型付き CPS を使用している。しかし、型付き A-normal form を使ったコンパイラの方がより良い変換をあたえることが分かっている。よって本稿では、型付きの A-normal form を用い、型を保存したコンパイラの設計と実装を行う。

論理学とプログラミング言語の間には以下のような関係がある。コンパイラのソース言語に用いる型付きラムダ計算と、論理学における自然演繹 (Natural Deduction) という体系は同一のものとみなすことができる。型付きラムダ計算と自然演繹における、型と命題、プログラムと証明は同じ性質を持つ。自然演繹において P が命題 A の証明ならば、型付き λ 計算において P が型 τ の値を計算するプログラムである。これをカーリー・ハワード同型性質 (Curry-Howard Isomorphism) [Curry:80][Howard:80][Gallier:93] と言う。

近年、この性質を用いることによって、プログラム変換を証明変換ととらえた分析がおこなわれてきた。型付き λ 計算から型付き A-normal form へのプログラム変換は、自然演繹から Genzen 流のシーケント計算の一種である GK [Kleene:52] への証明変換 (これを NG と呼ぶ) と、GK から GK のサブシステムである GKA への証明変換 (これを S と呼ぶ) の合成によって表される [ohori:99]。型付き A-normal form は、GK を A-normal 正規化した GKA と一致する。

コンパイラは最終的に逐次シーケント計算 (Sequential Sequent Calculus (SSC)) [ohori:99] と同じ性質を持つ論理抽象機械 SLAM へプログラム変換を行う。型付き A-normal form から SLAM へのプログラム変換は、GKA から逐次シーケント計算への証明変換と同じである。しかし、この証明変換はまだ定義されていない。

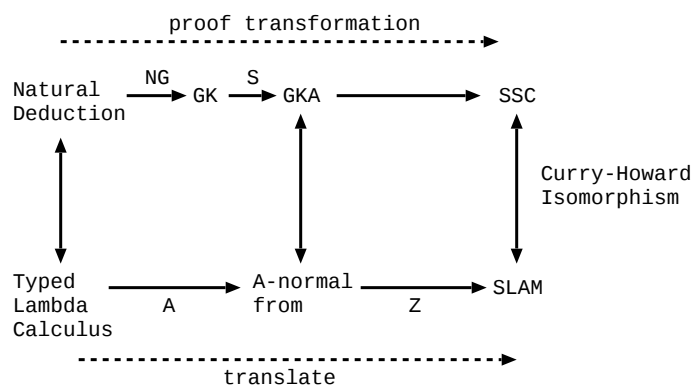


図 1: コンパイルにおけるカーリー・ハワード同型対応

プログラミング言語と論理学の対応関係について、図 1 に表す。

3 目的

本研究の目的は、A-normal form の論理的解釈を基礎に、A-normal form を中間言語とする型主導コンパイル方式を研究することである。実際には以下の 3 点を行う。まず既存の型理論の研究のコンパイラへの応用可能性を吟味する。次に A-normal form から SLAM へのプログラム変換における型の保存を証明する。最期に理論に基づいて実装を行なう。