

Title	コンパイル時自己反映計算を用いたカスタマイズ可能な言語処理系に関する研究
Author(s)	山田, 聖
Citation	
Issue Date	2000-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1337
Rights	
Description	Supervisor: 渡部 卓雄, 情報科学研究科, 修士



コンパイル時自己反映計算を用いた カスタマイズ可能な言語処理系に関する研究

山田 聖

北陸先端科学技術大学院大学 情報科学研究科

2000 年 2 月 15 日

キーワード： 自己反映計算, 型推論, Scheme.

1 目的

本研究の目的は、型推論機構を持つ動的型づけに基づく言語に対し、プログラムの構造に基づくパターンから型に関する情報を得るためのヒントとなる情報を、プログラマに明示させるための機構を提案することにある。更に、コンパイル時自己反映計算機構によって、このヒントとなる情報を明示するための機構を実現することにより、従来は構文木やパース木を扱う部分のカスタマイズのために使われていたコンパイル時自己反映計算が、その他の応用範囲に適用可能であることを示す。

2 背景

Scheme は動的な型づけに基づく言語である。このような言語では、実行時の値に対して型が決まる。変数は任意の型の値を束縛することができ、型宣言が不要であることから、プログラム記述の自由度が高く、プロトタイピングに適している。

C 言語や Standard ML のような静的に型づけされる言語では、全ての型検査を静的に行い、実行時プログラムを型安全で効率的なものにするのに対し、Scheme や Smalltalk のような動的に型づけされる言語では、本来の計算に加えて動的な型検査が実行時に行われるため、効率が悪く、また、型誤りの検出が難しい。これらの問題点を改善するために、Soft Typing や TypeRecovery といった、動的に型づけされる言語のための型推論機構が提案されている。これらは、定数の変数への束縛や原始関数の適用を元に、変数

に束縛される可能性のある値の集合としての型を推論するが、言語の持つ動的な側面のために保守的な結果しか得られない。記述性を損なうことなく、実行時コードの効率化、型誤りの早期検出を実現するためには、動的に型づけされる言語のための型推論機構が、コンパイル時により多くの型情報を得るための手法が必要となる。

3 本研究のアプローチ

動的型づけに基づく言語を用いる場合であっても、型を全く意識せずにプログラムを記述するようなケースはほとんど無いと言って良い。たとえば、原始関数に不適切な型の値を適用することを避けるためには、型に関する考慮を行いつつプログラムを記述しなければならない。このようなプログラマが持つ型に関する直観的な考えを明示するためのインターフェイスを用意すれば、それを用いることにより、コンパイル時により多くの型情報を得られるようになる。素朴な解決法として、静的な型づけに基づく言語に見られるような型宣言のための構文を用意することが思い付く。しかし、プログラマの持つ直観的な型に関する考えは、構文ではなくプログラムの論理的な構造に基づくため、この方法が適切であるとは言えない。

本研究では、構文とは異なった単位に基づいた型づけの規則をプログラマが明示するために、コンパイル時自己反映計算を用いた言語処理系を提案する。本研究のアプローチは、型推論機構が型を解析するフェーズを公開し、これをカスタマイズすることにより、柔軟な形式で型情報を記述するというものである。つまり、型推論機構の推論規則を修正、追加することにより、より多くの型情報が得られるようにする。

4 実現方法

本研究では、関数型言語 Lisp の方言の 1 つである Scheme のための型推論機構に対して、それが持つ推論規則をコンパイル時自己反映計算により拡張可能にする。

まず、Scheme のための型推論機構を操作的意味を用いて定義する。Scheme の本来の意味は、式と環境を元に値を計算する関数として表現されるのに対し、型推論機構に対する意味は、式、抽象実行時の環境、型環境から抽象的な値と新たな型環境を計算する関数として表現される。ここで、型環境とは、変数とそれが値を束縛する位置からその変数に束縛され得る抽象的な値の集合としての型への写像を指す。この型推論機構は、Scheme の本来の意味とは別の意味にもとづく解釈系とみなすことができる。

次に、ここで定義した型推論機構に対する自己反映計算を定義する。Scheme の式をベースレベル、その抽象実行を行う型推論機構をメタレベルとみなし、自己反映計算をメタレベルがベースレベルを解釈するための意味の変更としてとらえる。型推論機構に対する意味は操作的意味に基づく関数として定義したことから、意味の変更はこの意味関数の変更として定義される。具体的には、メタレベル記述として、構文の構造を指定す

るパターンと、その構文に応じた処理を表す手続きを与えることにより、意味関数の、ある構文からその構文のための処理を行う手続きへの写像が拡張される形となる。

この処理系に入力されるプログラムは、ベースレベルの記述である通常の式と、メタ記述から構成される。ベースレベルの式は構文木に変換され、型推論機構がそれを解釈実行する。型推論機構は、操作的意味の定義に従い、構文に応じて対応する計算を行うという処理を繰り返す。メタレベル記述が読み込まれると、それに基づき型推論機構に対し新たな構文とそれに対応する処理が追加される。その後読み込まれる式は、この拡張された型推論機構により解釈実行されることになる。

5 結論

本研究では、Scheme のための型推論機構に対する自己反映計算を用いたカスタマイズの方法を定義し、この方法に従い拡張可能な Scheme の型推論機構を実装した。この処理系を用いることにより、型推論機構による型の推論が困難な式に対して、型づけを行うことができるようになり、複雑な式に対する型の指定を柔軟に記述できるようになる。

また、このシステムを作成することにより、コンパイル時自己反映計算が、構文木やパース木等を扱うコンパイル処理の前段階のフェーズだけでなく、コンパイラの主要な処理の一つとなる型推論のフェーズをカスタマイズの対象とすることができることを示した。