

## 自己反映的な言語のモジュール結合による実装とその効率化

佐伯 豊<sup>†</sup> 渡部 卓雄<sup>†</sup>

自己反映的なプログラミング言語は、言語自身を計算の対象として扱うことが可能な言語システムである。

言語の実装レベルに相当する記述をメタレベルとして実際の計算に関する記述と分離することによって、実行時の状況に応じた動的な言語の拡張を、再利用可能なモジュールとして提供することが可能な枠組みであり、並列計算機や、分散計算機環境におけるアプリケーションの記述を簡潔におこなうことができる。

しかし、自己反映的な言語を実装する場合、言語の内部構造をユーザーにどのようにして提示するかを選択によって、その言語が提供する拡張性と実行効率とのトレードオフが生じる。

実際の自己反映的な言語では、並列計算など特定の用途に限った自己反映計算を提供する言語として設計されることが多く、実験的なものを除いては汎用的な用途に適した言語といえるようなものはない。本稿では、用途を限定しない自己反映的な言語の効率的な実装方法を提案したい。そこでそのような実装方法を提供するための枠組みとして、Monad-Transformer をもちいた。

本稿ではまず、自己反映的な言語のいくつかの形式を、拡張可能な機能ごとに独立したモジュールの組として定義するための枠組みを示し、次にプログラマーがプログラムの各局面に応じ、必要拡張機能と実行効率を考慮した言語の実装形態を随時選択するためのインターフェイスを提案する。最後に実際の言語システムの構築における問題点などに関する考察をおこなう。

Modular Implementation Technique for  
Efficient Reflective Programming LanguagesYUTAKA SAEKI<sup>†</sup> and TAKUO WATANABE<sup>†</sup>

Reflective programming languages are languages which enable user programs to run at the defining language level (meta-level), and act as a part of the interpreter. So reflective languages can provide high extensibility.

But in fact, in most of implementations of such a language, the given way of extension is not flexible enough. Because in the traditional methodology of defining reflective language is structuring the language as a meta-circular interpreter, so if we try to provide extensibility as possible. The performance to execute such programs are too low to build practical applications.

Some reflective systems restrict extensibility of the language, and solve meta-level computations at compile-time. But, if what we can do at meta-level is limited hardly, or extensions have effect on only statically, there is less advantage of reflection about extension.

In this paper, we propose a modular implementation technique of reflective languages. To provide the mechanism, we define some language building-blocks. And then we build a meta-continuation based reflective language as an interpreter and a compiler for the normal (non-reflective) language.

Our goal is designing interfaces to let the programmer possible to select the implementation of the reflective language and make user to be able to specify the language for the purpose of each program-pieces. So we conclude what is the most efficient combination of language modules for some domains.

(平成 11 年 6 月 17 日発表)

<sup>†</sup> 北陸先端科学技術大学院大学 情報科学研究科  
School of Information Science, Japan Advanced Institute of Science and Technology