
発表概要

自己反映的な言語における言語拡張同士の安全な結合について

佐 伯 豊[†] 渡 部 卓 雄[†]

自己反映的なプログラミング言語は、言語処理系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り、言語の実装レベルの記述を、メタレベルとして分離することによって、複雑なアプリケーションの記述を簡潔に行うことができる。また、メタレベルを上手にモジュール化することによって、拡張を再利用することができる。しかし言語拡張を再利用するためには、複数のメタのモジュール同士の意味的な衝突を考慮する必要がある。通常、再利用を前提として言語の拡張を記述する際には、その拡張を適用する対象の言語として純粋なものを想定することになる。そのような前提は、拡張同士を組み合わせることで保証できなくなる。そこで本稿では言語拡張同士の意味的な衝突を回避するための枠組みを示す。本稿では Monad Transformer を用いた言語システムのモジュール化を行い、言語を構成するコンポーネント間の依存関係を明確にする。その枠組みに基づいて、自己反映的なプログラミング言語を実装し、ある拡張と、それ以前の拡張との間で生じる衝突について検知し、拡張者に対してその情報を提供するための具体的な方法について提案する。

A Safe Composition Method for Reflective Extensions

YUTAKA SAEKI[†] and TAKUO WATANABE[†]

A reflective system can understand/compute its structure and behavior at run time. Giving the facility of reflection to OS or programming languages, it is possible to construct the advanced and complex systems, like mobile computing systems or distributed systems, with more systematic manner. Practically, the programmer might want to apply several extensions to the language simultaneously. But, there might be some semantical conflicts between these extensions. In this article, we propose a framework to avoid the conflicts. We provide a modular implementation of a reflective programming language based on monad transformers, and made its internal dependency of each component clear. And then, we propose a concrete mechanism to help a user who want to extend a language know enough informations to avoid possible conflicts.

(平成 12 年 3 月 23 日発表)

[†] 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