発表概要

強力でコンポーザブルな構文拡張機能をもつプログラミング言語

市川 和央[1,a)]　千葉 滋[1]

2013年8月2日発表

本発表では，新しいユーザ定義演算子とそれをサポートするプログラミング言語を提案する．我々の演算子はユーザ定義可能だが，構文の制限がほとんどない．演算子の構文はオペランドとオペレータの名前部分からなり，中置・前置・後置・外置以外の演算子も定義できる．我々の演算子は解析表現文法（PEG）と同等の表現力を持ち，実用的な言語で利用されている様々な構文を実現できる．我々の演算子のもう 1 つの長所はコンポーザビリティである．プログラマは演算子をモジュール化したライブラリを作成でき，またそれらのライブラリを複数組み合わせて利用することができる．たとえ複数のライブラリが同じ構文の演算子をもっていたとしても，それらは静的な型により区別される．このような演算子を実用的な時間で解析するために，我々は期待される型を利用した構文解析手法を開発した．構文解析器は次の式の期待される型の情報を利用して解析に利用する演算子を期待される型を返すもののみに制限する．型エラーになるような演算子は解析に利用されないため，効率的な解析が可能となっている．これと左再帰を許す packrat parsing を組み合わせることで，現実的な文法に対して線形時間で構文解析することができる．我々はこのユーザ定義演算子をサポートする Java 1.4 拡張言語を開発した．本言語は演算子を実装，利用するための簡単なモジュールシステムをもつ．また，我々の言語は演算子優先順位や結合性，サブタイプ関係を型情報の書き換えと演算子の追加により実現している．

A Programming Language Having Powerful and Composable Syntax Extensions

Kazuhiro Ichikawa[1,a)]　Shigeru Chiba[1]

Presented: August 2, 2013

In this presentation, we propose a new class of user-defined operators and a programming language supporting them. Our operators are user-definable, but the syntax of them is not restricted. The operator syntax consists of operands and operator name parts; they are not only infix, prefix, postfix, or outfix. Programmers can use them to implement various syntax found in practical languages since the expressiveness of our operators are equivalent to Parsing Expression Grammar (PEG). Another advantage of our operators is composability – programmers can implement a library for user-defined operators and use several libraries together. Even if some libraries have operators that have the same syntax, these operators are distinguished by static types. To parse these operators in pragmatic time, we developed a parsing method based on expected types. In this method, the parser uses expected type information of the next expression, and then tries parsing the input source expression assuming it is the operator that returns the expected type. Parsing is efficient since it prunes parsing paths for the operators returning unexpected types. It achieves O(n) parsing time against practical grammar by using packrat parsing supporting left recursion together. We have also developed a new language, which is an extension of Java 1.4 and supports our operators. It provides a simple module system to implement and export operators. Our language supports operator priority, operator associativity, and subtyping by rewriting type information and adding some operators automatically.

[1] 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo, Bunkyo, Tokyo 113–8656, Japan
[a)] ichikawa@csg.ci.i.u-tokyo.ac.jp