発表概要

Packrat Parsingを用いたRubyの構文解析

山口 大貴[1,a)]　前田 敦司[1]　山口 喜教[1]

2012年1月24日発表

多くのスクリプト言語同様，RubyはCやJavaといった従来の言語と比較して非常に複雑な文法を持っている．そのため，構文解析器の作成において現在主流となっている，yacc（またはbison）とlexを用いて構文解析器と字句解析器を生成しそれらを組み合わせる手法では，文法の記述や実装が困難であったり実装が複雑化しメンテナンスが困難となってしまったりする場合がある．たとえばRubyでは，文字列リテラルに任意の式を埋め込み可能な文法を実現するため等の理由で，手書きの字句解析器を含めて8000行以上にもおよぶ巨大な文法定義を行っている．これは，現在用いられている構文解析アルゴリズムが字句解析をベースとしているために，字句解析器に状態を付加する等のアドホックな実装を行わなければこのような文法を実現することができないためである．そこで本研究では，Parsing Expression Grammar（PEG）をベースとした，強力な解析力を持つ構文解析アルゴリズムPackrat ParsingをRuby処理系（JRuby）に導入することを提案する．Packrat Parsingを実際にRuby処理系に用いることで，従来の構文解析アルゴリズムで問題となっていた部分を改善し，文法定義の保守性を向上させることが本研究の目標である．本研究の構文解析器の実装は，PEGをベースとした文法定義をPackrat Parser生成系*Rats*!に与えることで行い，その文法定義は従来の文法定義を変換することで作成する．また提案手法による処理系と従来手法による処理系に対して実際のスクリプトを使用した比較評価を行い，その結果として，提案手法によって文法定義の保守性が向上することを示す．

A Packrat Parser for Ruby

Daiki Yamaguchi[1,a)]　Atsushi Maeda[1]　Yoshinori Yamaguchi[1]

Presented: January 24, 2012

Like many other scripting languages, Ruby programming language has relatively complex grammar when compared with more conventional languages such as Java or C. This complex grammar makes traditional parser implementation using yacc (or bison) and lex difficult, complicated, or hard to maintain. For example, in Ruby, over 8000 lines of grammar rule definition (including hand-written scanner) is used to implement complex token syntax such as string literals that can embed Ruby expressions. Since traditional parsers are based on token lookahead, they need some ad-hoc implementation technique such as stateful scanners. In this presentation, we propose a Ruby parser using Packrat Parsing, a powerful parsing algorithm based on Parsing Expression Grammar (PEG). We also show that it is possible to resolve the problems in traditional methods and increase maintainability of the grammar rule definition. We rewrite the definition of the grammar into PEG-based style from existing yacc-style one, and generate Packrat Parser by Rats!, a Packrat Parser generator. We also perform evaluation of our parser in comparison with existing Ruby parser using real programs. As a result, we show that our grammar rule definition is better than existing one in terms of maintainability.

[1] 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan
[a)] daiki@ialab.cs.tsukuba.ac.jp