

発表概要

マクロ処理の実装に適したバックラットパーサー コンビネータの設計と実装

今泉 良紀^{1,a)} 篠埜 功¹

2017年6月8日発表

バックラット構文解析では LALR(1) よりも強力な形式文法である Parsing Expression Grammar (以下 PEG) によって構文解析を行う。PEG による文法規則には曖昧さがなく、プログラミング言語の構文の記述やその処理系の作成に際し非常に有用である。既存の PEG ベースの構文解析器では直接文字列を入力として受け付けるが、字句解析と構文解析を分けたい場合にそのままでは使用に適さない。たとえば、字句ベースのマクロ処理系である C プリプロセッサを実装する際、マクロ展開後のコードに再度マクロ展開の処理が行われうる。その結果、マクロ展開の過程で同じレキシムに対して字句解析を複数回実行する場面がある。これを避けるために PEG ベースの構文解析器を使用しないとすると、字句解析や構文解析用に PEG 以外の形式文法を学んだりその処理系を導入したりする必要がある。そこで文字列だけでなく字句列などを入力できるバックラットパーサーコンビネータを C++ で実装し、本発表では実装の詳細について報告する。この実装を用いて生成した構文解析器は char 型だけでなく任意の型を返す前方反復子 (Forward Iterator) を入力として受け付ける。結果は tuple · variant · optional · expectedなどを組み合わせたデータ構造として得られ、意味規則によって自由に変換することが可能である。

Presentation Abstract

Design and Implementation of Packrat Parser Combinator for Implementing Macro Processor

YOSHIKI IMAIZUMI^{1,a)} ISAO SASANO¹

Presented: June 8, 2017

Packrat parsing is based on Parsing Expression Grammar (PEG), which is more expressive than LALR(1) and is not ambiguous. So it is useful on describing the syntax of a programming language and implementing the language. However, it is not suitable to use PEG-based parser generators when we separately implement lexical analysis and syntax analysis, since existing PEG-based parsers receive only a string. For example, since a C preprocessor, which is a token-based macro processor, may expand a code fragment that is obtained by expanding macros, a lexer may process same lexemes more than once in macro expansion. When we do not use PEG-based parser for avoiding unnecessary lexical analysis, we need to study and use another lexer and parser. In this presentation we report a packrat parser combinator, implemented in C++, that receives not only strings but also sequences of tokens. The parser described by the combinator can receive as its input a Forward Iterator that returns not only characters but also values of any type. The output of the parser is obtained as a data structure that consists of tuples, variants, optionals and expecteds. We can process the result by describing semantic actions.

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

¹ 芝浦工業大学大学院理工学研究所
Graduate School of Engineering and Science, Shibaura Institute of Technology, Koto, Tokyo 135-8548, Japan

a) ma17019@shibaura-it.ac.jp