

発表概要

解析表現文法に基づく C++ のパーサコンビネータ実装の 実行速度の比較

今泉 良紀^{1,a)} 篠埜 功¹

2018年2月28日発表

PEG (Parsing Expression Grammar, 解析表現文法) は文法規則に曖昧さのない形式文法であり, プログラミング言語の構文記述に有用である. PEG による文法は再帰下降構文解析器と対応しているため, パーサコンビネータによる簡易な実装によって PEG の文法から実際に動作するパーサが得られる. 特に EDSL (Embedded Domain Specific Language) によるパーサコンビネータ実装は, パーサの記述のし易さを確保しつつ, 意味規則の記述などの点でホスト言語との接続が簡潔であるという利点がある. また, バックトラックのある再帰下降構文解析器は素朴な実装では入力長に対して最悪指数関数時間を要するが, 再帰下降構文解析にメモ化を組み合わせたバックラット構文解析という手法によって入力長に対して線形時間で解析が可能となる. 本発表では, C++ で記述された PEG に基づくパーサコンビネータ実装として, Boost.Spirit.X3, cpp-peglib, PEGTL, matcha2 の 4 つの実装の実行性能について比較した結果と, これらの中で最も実行速度の速い matcha2 の実装の詳細について報告する.

Presentation Abstract

Comparison among the Runtime Performance of PEG-based Parser Combinator Implementations Written in C++

YOSHIKI IMAIZUMI^{1,a)} ISAO SASANO¹

Presented: February 28, 2018

PEGs (parsing expression grammars) are used for formally describing unambiguous grammars and thus are useful to describe the syntax of programming languages. Grammars described by PEGs are recognized by recursive descent parsers, which are obtained by combining parser combinators. By embedding combinators based on PEGs in some language, it becomes easy to construct parsers with semantic actions by having interoperability with the host language. Although it takes exponential time in the worst case to execute the parsers with backtracking in naive implementation, packrat parsers operate in linear time with respect to the input length by using memoization. In this presentation, we compare the runtime performance of four PEG-based parser combinators implemented in C++: Boost.Spirit.X3, cpp-peglib, PEGTL, and matcha2. Furthermore, we also report the implementation detail of matcha2, which runs fastest among them.

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

^{a)} ma17019@shibaura-it.ac.jp