

発表概要

メモ化表へのプリフェッチによる Packrat Parserの並列化

遠藤 仁^{†1} 前田 敦司^{†2} 山口 喜教^{†2}

(平成 23 年 4 月 25 日発表)

CPU の性能向上手法がクロックの高周波化からマルチコア化へとシフトした現在、CPU の性能を活用するために処理の並列化を図ることはますます重要性を増している。広く用いられているプログラミング言語の構文解析手法の多くは、有限の(多くの場合 1 つの)トークンを先読みすることによって逐次的に処理を進めるものであり、並列化は困難である。筆者らは、バックトラックによる再帰下降構文解析とメモ化を組み合わせた構文解析アルゴリズムである packrat parsing において、下位の非終端記号の解析結果を記録するメモ化表を、別のスレッドを用いてあらかじめ埋めておくことによって、上位の構文解析スレッドの動作を高速化できるのではないかと考えた。本研究では、Medeiros らの PEG 仮想マシンにメモ化機能を加えて packrat parsing 仮想マシンとし、さらに上記のアイデアを用いて並列化を行った。PEG で表記した文法を与えると、構文木を作成する並列 packrat parser を生成するパーザジェネレータを試作し、生成されたパーザに対して実際のプログラムを入力として評価実験を行い、並列処理の有効性を確認した。

Parallelizing Packrat Parsers with Memoization Table Prefetch

HITOSHI ENDOH,^{†1} ATUSI MAEDA^{†2}
and YOSHINORI YAMAGUCHI^{†2}

Parallelization of programs is getting more importance because recent performance improvement is driven mainly by increasing number of cores, rather than increase in clock frequency. Many of parsing algorithms used in programming language implementations rely on directing their actions by lookahead of finite (in many cases, only one) tokens, thus severely limiting the possibility of parallel processing. Packrat parsing is a variant of backtracking recursive descent parsing combined with memoization. In packrat parsing, memoization table for low-level nonterminals can be filled by distinct prefetch thread to ac-

celerate processing of higher-level nonterminals. Following the idea, we have parallelized packrat parsing abstract machine, which is based on PEG machine by Medeiros. We built a parser generator that generates parallel packrat parser from grammar description written in PEG. We evaluated performance of the generated parser using real program as input, confirming the effectiveness of our approach.

^{†1} 筑波大学情報科学類

College of Information Science, University of Tsukuba

^{†2} 筑波大学大学院情報システム工学研究科

Graduate School of Systems and Information Engineering, University of Tsukuba