

発表概要

限定継続を持つ並列言語モデルの設計と実装

田中 麻峰^{1,a)} 亀山 幸義¹

2011年7月29日発表

本研究では、純粋な関数型言語を拡張し、call/cc や例外などの逐次的なコントロールオペレータと、future などの並列計算プリミティブの2つを両立することのできる、適切な言語モデルの確立を目的としている。コントロールオペレータは、バックトラックやコルーチンなどの制御を表現でき、木構造データに対する様々なアルゴリズムや、ウェブアプリケーションの簡潔な記述が可能になるなど、大変有用である。一方、CPUのメニーコア化にともない、future プリミティブなどの明示的並列性を利用したアルゴリズムの効率化が重要となっている。本研究では、純粋な関数型言語を限定継続コントロールオペレータと、並列計算プリミティブにより拡張した言語を提案する。しかしながら、逐次的コントロールオペレータと計算の並列化とを併用すると、並列に行われる計算を逐次的にコントロールする、ということになり、そのまま組み合わせるとそれぞれの意味が容易に失われてしまう。この問題を解決するためには、並列化したプログラムが持つ意味が、それを並列化しないで実行した場合の結果と必ず同じであるように言語を設計する必要がある。そこで我々は、Moreauの先行研究をもとに、並列計算プリミティブと限定継続オペレータを併せ持ちながらも、逐次的意味が保たれるような言語をCPS変換を用いて実現した。

A Parallel Programming Language Model with Delimited Continuation

ASAMI TANAKA^{1,a)} YUKIYOSHI KAMEYAMA¹

Presented: July 29, 2011

Our goal is to establish a appropriate programming language model as a functional language with control operators and parallel primitives. Control operators are very useful because they allow us to represent controls such as back-tracking and coroutines, to write various search algorithms for trees and to support a compact description of web applications. The recent advance of many-core technology prompt us to parallelize more and more programs. While automatic parallelization is an attractive target for research, programming languages with explicit parallelization primitives such as future are of much interest. In this talk, we introduce a functional language which has delimited-control operators and parallel primitives. This combination of language primitives has been considered very difficult due to the conflict of fundamental notions. We solve this problem by applying Moreau's semantical framework to our setting, in which programs are guaranteed to have the same semantics as their sequential semantics.

¹ 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻

Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan

^{a)} asami@logic.cs.tsukuba.ac.jp