

Abstract

The lambda calculus since more than half a century is a model and foundation of functional programming languages. However, lambda expressions can be evaluated with different reduction strategies and thus, there is no fixed cost model nor one canonical implementation for all applications of the lambda calculus.

This dissertation is composed of four conference papers where: we present a systematic survey of reduction strategies of the lambda calculus; we take advantage of the functional correspondence as a tool for studying implementations of the lambda calculus by deriving an abstract machine for a precisely identified strong call-by-value reduction strategy; we improve it to obtain an efficient abstract machine for strong call by value and provide a time complexity analysis for the new machine with the use of a potential function; and we present the first provably efficient abstract machine for strong call by need.

Rachunek lambda od ponad pół wieku stanowi model i fundament funkcyjnych języków programowania. Jednak lambda wyrażenia mogą być wartościowane przez różne strategie redukcji i stąd nie ma ustalonego modelu kosztów ani kanonicznej implementacji dla wszystkich zastosowań rachunku lambda.

Niniejsza rozprawa składa się z czterech prac konferencyjnych, w których: prezentujemy systematyczny przegląd strategii redukcji w rachunku lambda; wykorzystujemy odpowiedniość funkcyjną jako narzędzie do studiowania implementacji rachunku lambda przez wyprowadzenie maszyny abstrakcyjnej dla precyzyjnie zidentyfikowanej strategii silnego wołania przez wartość; ulepszamy ją, by otrzymać wydajną maszynę abstrakcyjną dla tej samej strategii, i dostarczamy analizę złożoności czasowej dla nowej maszyny z zastosowaniem funkcji potencjału; prezentujemy pierwszą dowodliwie wydajną maszynę abstrakcyjną dla silnego wołania przez potrzebę.