

発表概要

コンパイラ実装言語で中間表現データ構造を記述するための言語機構

田畑 憲太^{1,a)} 上野 雄大^{2,b)} 大堀 淳^{2,c)}

2016年2月28日発表

本論文では、コンパイラ開発者がコンパイルフェイズを簡潔に実装するための言語機構を提案する。コンパイラでは、ユーザが入力したプログラムや変換の過程で経由する各中間表現をデータとして取り扱う必要がある。したがって、コンパイラ開発者はそれらをコンパイラ実装言語の一般的なデータ操作構文を通じて扱う必要がある。この操作構文は言語の構造との対応が自明ではないため、言語仕様記述に比べて多くのコードを書かなければならず、またその正しさも明確ではない。この問題に対する解決策として、本論文では、コンパイラ実装言語そのものをコンパイル対象のプログラムデータを表現する構文として使う機構を提案する。この言語機構は、コンパイラ実装言語で書かれたプログラム断片をデータとして扱う構文と、そのプログラム断片の解釈を与える構文からなる。この機構を使うことで、コンパイラ開発者はコンパイラフェイズが出力するプログラムの一部をコンパイラ実装言語の構文を用いて書くことができる。本論文では、バリエーションと組を含むラムダ計算に対して、この機構の拡張を加えた言語を定義し、その意味論を与える。さらに、Standard MLのサブセット言語のインタプリタにおけるこの機構の実装を報告する。またこの機構の実用型付き関数型言語への埋め込みについて議論する。

Approach to Writing an Intermediate Representation in a Language for Compiler Implementation

KENTA TABATA^{1,a)} KATSUHIRO UENO^{2,b)} ATSUSHI OHORI^{2,c)}

Presented: February 28, 2016

This presentation proposes a language extension that allows compiler implementors to write compilation phases concisely. This extension consists of two constructs: one that regards program fragments as user-level data and another that gives an interpretation to the fragments. These constructs enable compiler implementors to represent a part of output program as data directly in the language for compiler implementation. This presentation defines a lambda calculus extended with these constructs and presents its semantics. This presentation also reports an implementation of the extended calculus in an interpreter of a subset of Standard ML.

¹ 東北大学大学院情報科学研究科
Graduate School of Information Science, Tohoku University,
Sendai, Miyagi 980-8579, Japan

² 東北大学電気通信研究所
Research Institute of Electrical Communication, Tohoku
University, Sendai, Miyagi 980-8577, Japan

a) taba@riec.tohoku.ac.jp

b) katsu@riec.tohoku.ac.jp

c) ohori@riec.tohoku.ac.jp