
発表概要

Prolog へのラムダ項の導入とマクロとしての展開

井出 陽子^{†1} 向井 国昭^{†1}

本研究会では、HTML/JavaScript, Emacs, unix シェルなどの多言語との間の通信用として、Prolog をハブ言語として使うことを提案した。関数型言語の要素を導入し、そのために Prolog で関数型言語の汎用の評価器を試作した（情報処理学会プログラミング研究会発表資料 2008-1-6）。この発表では、その後の進展と応用を報告する。とくに、ラムダ記法をプログラムのロード時にコンピネータを介してマクロとして展開する方法を報告する。試験的な実験ではおよそ 60 倍の時間効率が得られている。さらに、拡張部分に対する自然なセマンティクスをロジックプログラミングの標準的な宣言的意味論の枠組みの中で与える。ロジックプログラミングへのラムダの導入は自然な発想であり、実際、先駆的な研究を含めいくつかの研究や実装がすでに存在する。できるだけ広くそれらと比較して 問題点を明らかにし、本研究の意義を明確にする。

Expanding Lambda Terms in Prolog as Macros

YOKO IDE^{†1} and KUNIAKI MUKAI^{†1}

We will present a report on continuing works on Prolog as a hub language between HTML/JavaScript, Emacs lisp, and unix shell (document number '2008-1-6' in the proceedings). Actually we introduced there functional expressions including lambda terms into Prolog and implemented evaluator for the extension. Here we describe in details how to expand lambda terms as macros using combinators as intermediate codes. Simplified pre experiments show that we can expect to get about 60 times better time efficiency than interpreter mode. Furthermore we give a natural semantics for the extension based on the standard declarative semantics of pure Prolog. Introducing lambda terms into logic programming seems so natural that there are already several studies and implementations. We will compare them and our work.

^{†1} 慶應義塾大学政策・メディア研究科
Graduate School of Media and Governance, Keio University