

㊦ Directions for Meta-Programming

J.W.Lloyd (Univ.of Bristol,英国)

発表要旨

Prologにおいてメタプログラミングを行う際の問題として、var, assert等のメタ述語の不純性と、メタレベルの変数とオブジェクトレベルの変数のおおのこのドメインが明確に区別されないことが挙げられる。

本論文では、次の(1)(2)の方法によって、この問題の解決を提案している。(1)メタレベルの項変数に(レベルを区別する)型を導入する。(2)オブジェクトレベルの変数/定数を表現するためのメタレベルの定数記号を用意し、groundな項でオブジェクトレベルの式/項を表現する。

発表では(2)について解説し、さらにgenerate & testによってminimal subprogramを見つける問題を例として述べた。

論文の後半はメタプログラミングの研究上の課題についてまとめられている。またこれからのよりよいロジックプログラミングのために、Prologの後継言語としては、declarativeなものであるべきだと提議があった。

質疑応答

質問：(インプリメント上、unifyがむずかしいという発表があったが)本当はunifyよりも(メタレベル)で作った代入のapplyを効率よく行うのがむずかしいのでは？

回答：unify, apply, renameは問題で、我々はまだ実際にこれらの問題には取り組んではいない。

質問：(オブジェクトレベルの項を、メタレベルで)関数として取り扱うのはどうだろうか。また我々はリフレクションを使うことがオブジェクト指向型プログラムのセマンティクスを与えるのに有効だと考えている。ロジックプログラミングの場合にも、こういったものを組み込みとすれば、システムプログラミングがきれいにできてよいのでは。

回答：それは正しいが、私はそういう方向はとらない。