

⑤ An Automatic Translation Scheme from Prolog
to the Andorra Kernel Language

F.Bueno*, M.V.Hermenegildo(Universidad Politécnica de Madrid,スペイン)

発表要旨

Andorra familyの言語 (Andorra Kernel Languagesを含む) は、基本的にはPrologのプログラミング・スタイルと、committed-choice型言語 (並列論理型言語) を同時にサポートすることを目的としている。ところが、AKLは、ユーザによるコントロールについて多少の詳細な記述を要求する。このことは、AKL上で走るPrologでプログラミングすることによって避けられる。しかしながら、PrologのプログラムはAKL上で直接実行することが出来ない。この要因は、多少の些細な統語上の違いから、cutの扱いやいくつかのタイプの並列性の抽出を可能にする事などの、より複雑な問題に至るまで数多く存在する。本論文は、PrologプログラムをAKLにcompileする自動コンパイラの構成に関する基本的なガイドラインを用意する。Prologをサポートすることに加えて、我々の変換方法は、可能な部分についてIndependent-AND並列の実行を成し遂げる。そしてそれは、この種の並列実行が変換を通して、元のPrologプログラムにおけるユーザが認める「複雑さ」を保存するという理由から適切である。

質疑応答

質問：変換できないプログラムは、どの様に扱うのか？ また、どの様な場合に、prologプログラムをAKLに変換できないのか？

回答：プログラムは全て変換できる。そして、そのプログラムは、cutによって維持される。cutが真にnoisyであるようなそれらの場合、それをquietとして実行できないので、我々は、noisy cut operatorを実行しないことが可能である。これらの目的の為に、あるoperatorをAKLにインプリメントした。それで、このoperatorを用いることにより、自動変換は、ほとんどの場合実行できる。そのoperatorがなければ、実行できない。

質問：なぜ、AKLを全く異なる技術であるSISCTusコンパイラと比較するのか？ 例えば、比較すべきはATコンパイラや最適化コンパイラであり、それに反して、AKLインプリメンテーションは一般に非常にナイーブなものだが。

回答：その点は明らかでない。あなたの言ったこの理由から一つのratioが存在し、また、AKLのインプリメンテーションは非常にナイーブなものであるゆえに、あるインプリメンテーションと他のもののインプリメンテーションに自然な違いが存在する。そして、そのratioはその違いの一つの典型である。そして、私が示してきた時間測定においてこのratioは考慮されている。一方、我々の目的であったAKLにハンドコード化したそれらのプログラムにおいて、我々は、それらの自動変換されたプログラムを比較できる。