

④ 制約プログラミングシステム

—CAL, GDCCとその制約評価系—

(発表者：ICOT研究所 第4研究室 相場 亮, 長谷川 隆三)

質問：インペリアル大学のKeith Clarkです。多重環境について関心を持っているのですが、インプリメンテーションにはどのくらいのコストがかかったでしょうか。また、どのくらいのレベルでインタープリテーションを行っているのでしょうか。つまり、組み込みのユニフィケーションではなく、自分でユニフィケーションを行っているのでしょうか。

回答：GDCCの現在のバージョンは、ほんとうに実験的なバージョンで最初のバージョンです。したがって、実際にはKL1のShoenメカニズムを使ってブロック構造を実現しています。GDCCのブロック構造は、Shoen KL1オブジェクトプログラムではShoenにコンパイルされます。多くの実験を行った結果を正直に言いますと、そのようなインプリメンテーションはそれほど効率的ではありません。そこで、もっと効率的で軽いインプリメンテーション方法を考えて行かなければならないと考えています。

質問：Shoenは多重束縛環境をサポートしていることはわかります。変数に対する他の束縛はどのように行なっているのでしょうか。

回答：もともとShoenをKL1に導入したのは、並列推論マシン上にオペレーティングシステムを実現するためでした。したがって、失敗を局所化する機能などを持っています。多重環

境の処理に関しては、変数に対していくつかの内部表現を導入しています。したがって、異なる値をもつ同じ変数は、実際には内部的には異なった形式で表現されています。

質問：Thierry Le Provoostです。ドイツのECRCから参りました。逐次型制約論理プログラム言語CALからGDCCに並列化、あるいは改良をされていますね。また、非決定的アプローチのCLPスキーマからCCフレームワークへと変更されていますが、その動機は为什么呢。つまり、より優れた並列性のためでしょうか、それとも他に目的があるのでしょうか。

回答：CALを使っていくつかのプログラムを書いた経験から申しますと、そのようなプログラムの多くは、それほど非決定的であるとは言えません。ほとんどのプログラムが決定的プログラムであり、最初は単に制約の概念をコミテッドチョイス言語(committed choice language)に導入して並列化してみようとしたわけです。しかしその後、興味深い、そして重要なアプリケーション領域で、探索を行う必要があるものをいくつかみつけました。そこで、コミテッドチョイス言語にブロックメカニズムを組み込む必要があったわけです。