

④ Multiple Reasoning Styles in Logic Programming

H.Gallaire(ECRC,西独)

発表要旨

論理プログラミングは産業界に浸透する必要がある。もし論理プログラミングがたくさんのreasoning styleを支援するツールであるなら、その浸透のスピードは速まるだろう。ここでは、いくつかのreasoning styleについて述べ、さらに論理プログラミングがより完全にそれらのstyleを支援していくための拡張について述べる。この拡張は制約プログラミング(constraint programming)と呼ばれるものである。拡張を支援するメカニズムの非常に密接な統合というものを肯定する立場でここでは論じる。そのような立場をとるためには論理プログラミングの核の部分の実装を、根底から変更する必要があるときがあるが、なぜそうなのかについてもここでは触れる。この変更の作業が実現困難なほどでは無いことがCHIP,CLP,PrologIII,Trilogyといったシステムによって示されているといえる。

質疑応答

質問：logic programmingにおいて論理を拡張(extended logic programming)したり、あるいはfunctionを取り込んだり(functional logic programming)して機能を拡張するという試みを、自分のところでも研究していてそれなりに成果を出している。

回答：確かにそうやってsemanticsを変えていくという方向は大事なことだと思っている。ただ複雑なことになるが。

質問：現在の研究の大きなテーマとして並列処理というのが1つあると思うがその並列処理とlogic programmingの関係について何かアイディアはあるか。

回答：確かに並列処理は処理スピードが上がるし面白い研究分野だと思う。logic programmingは並列探索, OR並列に適していると思う。