

「関数型言語向きマシン」小特集の編集にあたって

東 田 正 信†

近年、エキスパートシステム、推論マシンなどの知能処理の研究が精力的に進められており、この分野に適した知能処理記述言語の議論も盛んになっている。LISP に代表される関数型言語は、プログラム記述の容易なこと、高度な並列処理が可能なことなどから注目されている言語の一つである。

関数型言語の起源は1930年代にA. チャーチらによって論理的・数学的な基礎が確立されたラムダ計算法(Lambda calculus)にまでさかのぼる。その後、1960年にJ. マッカーシーによってラムダ記法にもとづくLISP言語が開発された。さらに、1968年から1975年にかけて、ソフトウェアの書きやすさや並列処理による高速化を目的とした関数型言語やリダクション言語が開発された。

一方、関数型言語の特徴を活かし高度な並列処理を直接実行できるような計算モデル、処理系の研究も1960年代から積極的に進められてきた。その代表的な例がデータフローマシン、リダクションマシンである。これらのマシンで採用されているアーキテクチャは、バッカスによって提起されたフォンノイマンボトルネックを解消するいわゆる非ノイマン型のものである。これらのマシンの大きな特徴は、従来のノイマン型のデータ処理方式がコントロールフローに従ってデータ値が参照されていたのに対し、リダクション型では可能な個所から式を簡略化(リダクション)することによってデータ処理を進めて行く方式を採用しており、データフロー型ではデータそのものの流れがデータ処理の制御を行っていく方式を採用していることである。

これらの研究はLSI技術の発達に伴い関数型プログラムの直接実行が現実的な速度で可能となったこと、さらに1980年代に入って第五世代計算機の構想

が発表され、知能処理での並列処理の高度化・高速化の検討の必要性が高まったことなどから一層活発になってきた。

このような状況を背景に、本小特集では「関数型言語向きマシン」と題して、関数型プログラムを高速に実行できる専用マシンの研究・開発動向を紹介することを目的に、リダクションマシン、データフローマシンに関して3篇の解説を掲載した。

第1章では、リダクションマシンに関して、リダクションの原理から、処理方式、マシンアーキテクチャまでを試作例などを含めて解説をした。

第2章、第3章では、データフローマシンをリスト処理指向のものと数値処理指向のものに分け、それぞれの処理方式、高速化の手法、ハードウェア構造の特徴などについて解説をした。

なお、はじめてこのような関係の解説を読まれる読者のために、第1章、第2章の前半のかなりのページを割いて、関数型言語の特徴、それぞれの計算の原理などについて詳細に解説した。

リダクションマシンもデータフローマシンも理論面での検討は古くから行われているが、実際にハードウェアの試作・評価がされるようになったのは1970年代の後半になってからであり、現在はマシンの実用性・適用領域を含めた本格的な研究段階に入っている。このように非ノイマン型マシンの研究は、現段階では従来のノイマン型マシンを置き換える程の成果は得られていないものの、技術蓄積は着実に進められており、将来の計算機アーキテクチャの検討に対して大きなインパクトを与えることが期待されている。

最後に、ご多忙中にもかかわらず、執筆を快く引き受けて下さった執筆者の方々、ならびにご査読いただいた方々に厚くお礼申し上げます。

(昭和60年5月28日)

† NTT 技術企画本部