

特集「関数型プログラミングとその応用」の編集にあたって

清 木 康†

関数型プログラミングは、1930年代に A. Church によって確立されたラムダ計算 (Lambda Calculus) を起源とし、1960年代初頭に、J. McCarthy によって設計された関数型言語 pure Lisp の処理系の実現を経て、計算機上で実現可能なプログラミングの方法論として注目されるようになった。そして、1978年に、J. Backus によって行われたチューリング賞記念講演を契機に、関数型プログラミングに関する研究はさらに活発となり、この10年間に数多くの関数型言語、処理系、計算機アーキテクチャが提案されてきた。

手続き型言語によるプログラミングとの対比において、関数型プログラミングは、代入の概念がないという性質、すなわち、参照透明性 (referential transparency) によって特徴づけられる。この性質により、関数型プログラミングは、ソフトウェアおよび計算機アーキテクチャの両面から注目を集めてきた。ソフトウェアの側からは、手続き型言語によるプログラミングにおいて指摘されてきたプログラムの生産性の問題、すなわち、ソフトウェア危機を解決する可能性が注目されてきた。また、計算機アーキテクチャの側からは、計算の実行順序がデータの生産と消費の依存関係だけに制約されるという性質により、任意の関数型プログラムに内在する並列性の抽出が容易となる特徴が注目されてきた。そして、並列処理アーキテクチャを設計する上での基本的な計算方式として、関数型プログラミングの概念の適用の可能性が研究されてきた。

このような背景の中で、現在、関数型プログラミングの概念は、計算機科学の分野に多大なる影響を与えている。そこで、関数型プログラミングの概念、およびそれに関する研究をここでもう一度整理し、この分野の研究に関する今後の方向づけを考えるために、本特集では、関数型プログラミングに関する理論、言語、計算機アーキテクチャ、そして応用に関する研究の成果をまとめる。

本特集は、4部から構成され、全11編の解説を含んでいる。

第1部は、関数型言語とその計算モデルに関する5編の解説から成る。ここでは、関数型プログラミングの概念の成立の背景、およびその計算モデルについてまとめる。まず、関数型言語の基本的な概念と実際の関数型言語 (横内)、計算系と論理体系からみた計算モデルに関する成果 (富樫)、計算モデルの一つとして知られる項書き換え系に関する意味論 (稲垣、直井)、について解説する。次に、関数型プログラムと高い親和性をつストリームに関する計算 (田中)、および、最近、特に新たな進展が見られるデータ型の理論 (林) について解説する。

第2部では、関数型プログラムの実現方式に関する技術として、アーキテクチャ技術、リダクション技術 (小長谷、山本)、および、コンパイル時における最適化技術とその実現技術 (小野) について解説する。ここでは、関数型プログラムの実行方式と手続き型言語によるプログラムのそれとの相違点、および関数型プログラムと並列処理の親和性についてまとめている。

第3部では、関数型プログラムの作成支援およびデバッグに関する技術 (高橋)、また、プログラムの仕様記述から関数型プログラムを得、それを実行するまでの一連の処理系 (東野、関、谷口) について解説する。

第4部では、関数型プログラミングの概念の応用について述べる。ここでは、計算機システムの主要な機能の一つとなったデータベースの分野への関数型プログラミングの概念の適用 (清木、加藤)、および、計算機システムの重要な応用となっている画像処理分野への関数型プログラミングの概念の適用 (太田) について解説する。

最後に、ご多忙中にもかかわらず、執筆を引き受けてくださった著者の方々、ならびにご査読の労をとおめくださった方々に厚くお礼申し上げます。

(昭和63年7月11日)

† 筑波大学電子情報工学会