

■胡振江 アーキテクチャ科学研究系 教授

### 【タイトル】

**プログラミングを科学して、効率や信頼性を高める**

### 【本文】

私の研究テーマを一言で言うと、「プログラミングを科学する」ということです。例えば、小学生が鶴亀算を解くには、場合を列挙していかなくてはなりません、中学生になって変数を用いて方程式を立てるというルールを教えてもらえば、あと演算だけですむようになります。同じように、これまで経験が頼みだったプログラミングについても、その背景にある数理的構造などからプログラムを演算（演算）するための理論を構築することにより、それをマスターした人には容易に、信頼性が高く安全なプログラムを効率的に作れるようにしようというものです。

こうした理論によれば、効率を念頭に置いてプログラムをしなくても、単一のコンピューターだけで処理を行う逐次プログラミングから、複数のコンピューターを協調させて同時に制御する並列プログラミングへと容易に変換することができるようになります。また、例えば、ウェブページを作成するときには、通常はデータを入力することでページが生成されますが、逆にページのほうに変更を加えた場合には、元のデータも自動的に変更できるという“逆変換”のプログラムも作れるようになります。

### 並列プログラムを4つのブロックだけで構成

近年、ハードウェアの高性能化と低価格化が進んだため、大規模並列計算が身近なものとなっています。このため、理論研究の業績を基にして、特に並列プログラミングについて企業と連携して実用化を目指しています。一般の人にも開発しやすいように、「スケルトン並列プログラミング」という手法を提案しました。並列プログラムを、頻出する計算パターンを抽象化したブロック（並列スケルトン）に分解できれば、まるで積み木を組み立てるような感覚でプログラムが作れるようになります。われわれは、基本となる4種類のブロックがあれば、それらを自在に組み合わせるだけで、ほとんどの並列プログラムが作成可能なことを実証しました。プログラミング言語の中にこうした並列プログラミングの手法を組み込んでおけば、自動的に4種類のブロックに分割して、効率的に組み立てられるようになります。

### 一貫性を保ってプログラムを発展させる双方向変換

もう1つ、短期的な研究テーマとして、双方向変換の実用化に挑んでいます。大規模プログラム開発では、ユーザーの仕様を受けて、設計、記述（コーディング）、テストというように、各段階で別の人が携わっています。プログラムを進化させようとして、それぞれの段階で変更点が生じた際に、“逆変換”によって、その前の段階にもさかのぼって変更を反映させられれば、システムの一貫性を保証できるようになります。

社会におけるソフトウェアの重要性は、ますます拡大する一方です。信頼性の低いものが世の中に送り出されれば、かつての 2000 年問題のように、負の遺産が残ることにもなりかねません。科学に裏打ちされた理論をベースとして備えて、資質を高めた人がプログラミングにかかわっていくのは、とても大切なことです。

(取材・構成 塚崎朝子)