

特集「並行処理におけるプログラム試験」の編集にあたって

古川 善吾¹

¹九州大学情報処理教育センター

並行処理が実用に供されるにしたがってその試験方法の確立が必要になってきている。インターネットに代表される計算機ネットワークや並列計算機の普及によって並列・分散システムの利用が広がっている。そのために、これら並列・分散システムに障害が発生するとその影響が大きくなる。システムの障害を減らすための現実的な方法がソフトウェア試験である。

並行処理は計算機科学の中で何度も登場する概念であるけれども、用語さえも統一されていない。ここでは、「並列」を並列計算機というように、ハードウェアとしての並列性を表すために用い、「分散」は分散協調システムというように異なった機能をもった計算機あるいはプログラムが連携することを表す。並列や分散がハードウェアやプログラムという具体的なものと結びつきが強い概念であるのに対し、並行は、それぞれ独立に動作している処理が必要に応じて相互作用を行うことを表す抽象的な概念である。本特集では、上記のような抽象的な概念としての並行処理に対する試験方法を解説する。

実際に試験する対象はプログラムである。プログラムは、ハードウェアによって実行され、試験によって誤りを発見する対象となる。この「誤り」は、プログラムの仕様や使い方に対してプログラムがそのとおりに動作しないことと定義される。仕様や使い方まで含める場合にはソフトウェアという呼び方をする。ソフトウェア試験技法 (testing) は、仕様に対するプログラムの誤りを発見するための技法である。

試験は、プログラムが仕様通りに動作しない不具合 (failure) の発見を目的としている。一方、不具合が発生する原因としてプログラムのどこに障害 (fault) があるか、どう直せばよいかについては、デバッグ技法 (debugging) として議論されている。試験は、プログラムの誤りを発見して回避、除去することによって信頼性を高める方法である。一方、検証 (verification) は、形式的な技法を用いてプログラムの正当性を積極的に証明しようとする試み (広い意味では、プログラムの静的な解析や試験を含めた、信頼性向上のための技法を検証と呼ぶことがある) である。検証は、高価であり、大規模なプログラム全体に適用することは困難であるので、実用的な技法としてはソフトウェア試験が用いられてきている。

試験には、(1) 効率化、(2) 高品質化、という課題がある。1973年に開催されたソフトウェア試験に関する最初の国際会議では、主に試験の効率化が議論された。その後、1975年に、GoodenoughとGerhertが試験項目の品質について形式的な議論を始めた。試験によってプログラムの正当性を示すためには、すべての入力データを試験する「全数試験 (exhaustive testing)」が必要であるけれども、現実的な方法ではない。そのために試験すべき項目についての条件を定める試験基準 (testing criteria) が提案され、実際の試験において試験基準を満足した時の信頼性について議論がなされている。並行処理プログラムの試験方法については、1986年の試験に関するワークショップでのTaylorの提案以来議論されている。

試験の品質を計る方法として、上記の試験基準だけでなく、ソフトウェアの開発における障害の発見過程を用いた品質評価技法がある。これまでのソフトウェアの信頼性の算出は、開発方法を調べることによって行われてきた。これに対して最近提案されている「統計的試験方法」は、利用方法に基づいて、ソフトウェアそのものの信頼性を求めるための試験方法である。

本特集では、最近身近になってきている並行処理におけるプログラム試験に関して以下の4編で解説する。

第1編「並行処理プログラムの試験」では、並行処理プログラムの特性とその試験方法について説明する。第2編「通信プログラムの試験」では、並行処理として並列に運用されている通信システム間での通信を確認するための相互運用試験について説明する。同時に、通信システムと仕様との不整合を検出するための適合性試験についても説明する。第3編「制御用プログラムの試験」では、制御用プログラムの特徴およびその試験を概説する。さらに、制御用プログラムの試験で不可欠な制御対象シミュレータの作成支援についても説明する。第4編「クライアント・サーバプログラムの試験」では、分散処理の代表的な方式であるクライアント・サーバプログラムについて実際の開発現場における試験方法を紹介する。

最後に、ご多忙中にもかかわらず快く執筆を引き受けてくださった著者の皆様ならびに貴重なコメントをいただきました読者の皆様に感謝いたします。

(平成9年10月2日)