

刊行主旨

情報科学に使われている数学は、非常に幅広く、また奥深い。さらには、情報科学から発展して数学の一部門を形成しつつある分野もある。そのため、現在の大学教育の中で「情報科学のための数学」を体系化することは、まだ十分にできているとはいえない。それは、数学のさまざまな分野から情報科学に役立つものを摘出し、限られた期間内に効果的に学生に提供することの困難さと、情報科学が新しい学問であり流動的であるだけに、その中で役立つ数学を整理し統合化しようというには、まだ時期尚早なためでもある。しかし、情報科学がしっかりと地につき発展していくためには、どの時点においても数学との協調が必要不可欠である。

このような視点のもとに、現時点での「情報科学のための数学」を大胆に整理・統合化し、本「情報数学講座（全15巻）」を構成した。そのため、ある分野は重点的にとりあげ、ある分野は縮小・併合することになった。たとえば、数値解析が省かれているが、これはそれだけで一つのシリーズを形成するほど大きな分野であることもあって、本講座では他にゆだねることにした。

各巻の内容については、次のことを目標に定めた。

- ① 「役に立つ数学」を提供することをモットーとする。
- ② 定理・証明の羅列は控え、話題の展開や表現の仕方を工夫する。
- ③ コンピュータ・サイエンスからの具体例を豊富に入れ、抽象的概念をわかりやすく解説する。
- ④ 「その彼方に何があるのか」、「その分野が何を目指しているのか」に

ii 刊行主旨

言及し，筆者のフィロソフィを重視している．

本講座が情報科学の発展の一助になれば幸いである．

編集委員

まえがき

デジタルコンピュータを中核とする情報処理システムは、それ以前のアナログ機械と違い、複雑で大規模な問題も解くことが可能な、汎用の問題解決システムである。その基本原理は、問題解決に必要な情報や知識などをコード化、すなわち記号の列で表現し、その記号列を機械によって解釈し変換する、というものである。この問題解決パラダイムは、人が、記号の集合である文字（アルファベット）で構成された言語によって知的な活動を行うことに対比しうる、一般性を持つものである。この記号列として表現された情報や処理手順がプログラムである。

簡単な処理を行うプログラムであれば、コンピュータで直接実行可能な、機械命令の列として書くことも不可能ではないが、複雑で大規模な問題を解決するためには、より高水準のプログラムの記述システム、すなわち**プログラミング言語**が必要となる。自然言語が、情報の伝達手段にとどまらず、人間の思考活動の枠組みとしての役割を果たしているのと同様に、プログラミング言語も、コンピュータへ動作を指示するための手段にとどまらず、複雑で大規模なソフトウェアを構築するための枠組みを与える重要なものである。数学を典型とする近代の科学が、種々の抽象概念を組織的かつ階層的に導入することを可能にする記述システムを持つことによって急速な発展を遂げたように、ソフトウェア科学・工学が、複雑なソフトウェアを信頼性を持って効率よく開発する体系を確立し発展させていくためには、ソフトウェア構築上有用な種々の抽象化の概念や構造化の機構を効率よく構築していくことを可能にする、プログラムの記述システムが必要である。近代的なプログラミング言語の主な役割は、そのような記述システムを提供することである。

今日のソフトウェアシステムは、種々の形態のメディアおよび新しい計算パラダイムの出現により、急速にその複雑さを増しつつある。それに伴い、

iv ま え が き

情報処理の種々の分野において、より高度な機能を装備した新しいプログラミング言語の開発が望まれている。巨大な論理体系である大規模なソフトウェアシステムを記述する基礎となるプログラミング言語は、種々の豊富な機能を提供すると同時に、その言語で表現可能ないかなるプログラムの意味も厳密に定義されている、整合性ある体系でなければならない。そのようなプログラミング言語を実現するためには、プログラミング言語の持つ諸機能の間の相互関係の理論的分析や整合性の検証、およびそれに基づく厳密な意味の定義等が必要である。本書のテーマであるプログラミング言語の基礎理論の目的は、そのような分析や検証を可能にし、実用的なプログラミング言語の設計および実装の基礎を与えることである。

理論的な基礎に基づき開発された実用プログラミング言語の（数少ない）具体例として、ML 言語を挙げることができる。ML の定義 [38] は、誰が読んでも唯一の意味を持つ型理論の概念を用いて書かれており、その実装は、言語の意味の定義に基づき系統的に行われている。この厳密性は、ML で書かれたプログラムに極めて高い信頼性を与えている。さらに、ML は、多相型関数や型の自動推論機構などの高度な機能を提供している。これらの機能は複雑なソフトウェアを信頼性を持ってしかも効率よく開発していく上で有効性が高いが、これらは理論的基礎に基づく系統的な設計によって初めて可能になったものである。以上のような特徴を持つ ML (Standard ML) は、現時点では、最も良く設計された高水準の実用プログラミング言語の一つであり、またプログラミング言語の新しい機能等の研究の基礎ともなっている言語である。

この例からも伺われる通り、プログラミング言語の理論的基礎は、単なる理論的な興味の対象ではなく、望ましい機能を持つ実用プログラミング言語の系統的な開発を可能にする鍵となるものでもある。本書の目的は、プログラミング言語の数理的モデルを理解し、プログラミング言語の理論の基礎を習得することである。特に、ML を典型とする近代的なプログラミング言語の動作原理を理解するための基礎知識を獲得することを、具体的な目標とする。

プログラミング言語の理論は、**構文論** (syntax) と **意味論** (semantics) に

大別される。構文論は、文としてのプログラムそのものの性質を扱う理論であり、プログラムの文法構造やその構成要素の持つべき型に関する性質、およびプログラムの同値性、プログラムが具体的にどのような操作を表現するか、等を対象とする。プログラムの表現する操作は、プログラムの意味にかかわることであるが、プログラムの文法構造に則して具体的に定義される意味は、プログラムの**操作的意味論** (operational semantics) と呼ばれ、プログラムの構文論の一部として論じられる。これに対してプログラムの意味論は、プログラムが表現 (表示) する計算そのものを、プログラムの文法的な構造に依存しない抽象的数学的対象として表現し、それを通じてプログラムの持つ性質を調べる理論であり、より厳密には**表示的意味論** (denotational semantics) と呼ばれる。

これらのなかで、プログラミング言語の設計や実装の直接的な基礎となるものは、構文論的諸性質、特にプログラムの持ちうる型に関する性質、プログラムの操作的意味論、およびその両者の関係である。そこで、本書では、プログラミング言語の構文論的性質を詳しく取り扱う。プログラミング言語の表示的意味論については、その一般的な枠組み、およびその構文論との関係等を主に取り扱う。プログラミング言語の具体的な表示的意味論 (モデル) の構築も、種々の興味深い問題を含んだ重要な研究対象であるが、このテーマについては、例えば文献 [54, 63, 49, 26, 64] などの教科書で、すでに詳しく取り扱われているので、詳しい内容はそれら良書に譲ることにする。

以下第 1 章で、本書で取り扱う基礎理論の構造、および他の理論との関連について説明した後、次章以降で展開するプログラミング言語の基礎理論の準備として、形式言語の帰納的な定義、およびラムダ計算に基づく計算モデルを概説する。さらに、その後の章で扱う種々の型システムの構造およびそれ以降の本書の展開について説明する。

本書は、情報科学関連の学部や大学院初年度のレベルを念頭においているが、プログラミング言語の基礎に興味のある者であれば、情報科学の基礎知識がなくても理解できるよう配慮し、必要な概念はすべて定義するように心がけた。そのような本書の性格上、取り上げることができなかった重要な概念や手法が数多くある。より深い理解のためには、上に挙げた表示的意味

vi ま え が き

論の教科書の他に，本書のテーマに関連の深い計算モデルに関する教科書 [60, 62] や，型理論が詳しく扱われている教科書 [40]などを参考にされることを勧める．また，より専門的な学習の手がかりとして，本文中になるべく多くの関連する論文の引用を含めた．

最後に，本書の草稿を一緒にセミナーで読み，種々の誤り等を指摘下さった加藤岳臣氏，田村宏樹氏，および草稿を注意深く読んで頂いた南出靖彦氏，小林孝次郎先生，本書の執筆をすすめて下さった武市正人先生，本書執筆のあいだお世話になった共立出版株式会社の坂野一寿氏に深謝いたします．

1997年1月

大堀 淳