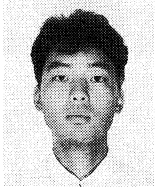


している技術者が考えを整理する手助けとして用いても十分に活用できるであろう。高い評価を得た前著に続く本であり、また最近の分散 OS に対する需要を反映した内容となっているため待望の一冊といえる。



八十田 暁太 (正会員)

1990年慶應義塾大学理工学部電気工学科卒業。同年より三菱電機情報電子研究所勤務。現在、分散システム構築技術の研究開発に従事。

B. A. Davey and H. A. Priestley 著

## “Introduction to Lattices and Order”

Cambridge University Press, 248p., 1990

本書は、計算理論の基礎となる順序集合と束に関する教科書であり、基本理論と豊富な応用例を含んでいる。数学科の学生と論理学、離散数学、計算機科学など隣接分野の研究者を対象とする。

本書は、11章から構成される。第1, 2章, 第5章から第8章が本書の中心で、順序集合と束に関する基礎を提供する。また、第3, 4章がプログラミング言語の表示的意味論の数学的背景となる Scott の情報システムと不動点定理を取り扱っていることから、第1章から第4章で計算理論のための基礎を提供していると言える。

第1章「順序集合」は、数学、計算機科学、社会科学からの具体的な例を提示しながら、半順序集合に関する基本概念を説明する。

第2章「束と完備束」は、順序集合の特殊なものとして束と完備束を定義する。そして、閉包演算子を介した完備束と集合積構造の関係、完備化などについて説明する。

第3章「CPO, 代数束とドメイン」は、代数束, Scott 理論のドメインと情報システム, 不動点定理で用いられる CPO (complete partially ordered set) などを結び付ける。応用として簡単にプログラミング言語の表示的意味論を紹介する。

第4章「不動点定理」は、不動点定理とその応用について説明した後、順序論の概念である極大性公理と集合論の選択公理を結ぶ Zorn の補題などを説明する。この章は前章の議論が基盤になっているが、前章を理解できなくても、不動点定理を理解することはできる。

第5章「代数構造としての束」からは、一転して束に関する公理から出発する代数的説明が中心となる。第3章と第4章とは直接的に関係しないので、第2章から本章へともでも問題はない。この章では、束を代数的に定義し、合同や商束、束に関する準同形定理など普遍代数的概念を説明する。

第6章「モジュラ束と分配束」と第7章「ブール束と応用」では、特殊な束として、モジュラ束と分配束, ブール束 (ブール代数) を導入する。第7章では、ブール代数を定義した後、計算機回路や演繹システムなど論理学との関連を説明する。

第8章「表現理論: 有限」と第10章「表現理論: 一般」は、表現理論に関するもので、分配束, ブール束を集合の包含関係を用いて表す方法を示す。第8章では有限の分配束とブール束を、第10章では無限集合を含む場合の取り扱い方法を説明する。これらの間の第9章「イデアルとフィルタ」では、命題計算なども関連したイデアルとフィルタを説明する。

第11章「形式概念分析」では、応用として社会科学などで用いられる概念分析の基礎を説明する。

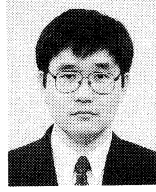
本書は、順序と束に関して広範、かつ、網羅的に扱っている。この網羅性が記述の平板化につながり、ポイントが把握しにくい面もある。評者はこの分野が専門ではないが、知る限りの他の束に関する教科書、例えば、Birkhoff<sup>1)</sup> と比べると、計算機科学と関連した話題が多く取り上げられている。また、具体例や図が多いため、イメージをつかみやすい。ただ、第3章だけは具体例や図が少なく、多くの概念が導入されるため、この分野に不慣れな読者を混乱させるかも知れないし、何度か頁を歩きつ戻りつさせることになるであろう。本書を途中で投げ出すとしたら、この章が一番の候補であろう。評者らは本書の前半を輪講したが、この章に一番時間を費やした。

本書は、プログラミング言語の表示的意味論に関して、数学的背景を勉強したい読者に勧められる。評者の周囲には、本書で不動点定理を理解したので、最小エルブランモデルが簡単に理解できたという人もいる。また、電子ニュース中で、プログラミング言語の意味論の背景になる数学の

教科書として本書を推奨している人もいる。ただし、少しハードであるが、また、束を詳細に勉強したい読者、最近の計算言語理論で用いられている素性構造の数学的背景を勉強したい読者には良い教科書となるかも知れない。

### 参考文献

- 1) Garrett Birkhoff. *Lattice Theory*. American Mathematical Society, Providence, Rhode Island, USA, 3rd edition (1967).



小暮 潔 (正会員)

1979年慶應義塾大学工学部卒業。  
1981年同大学院修士課程修了。同年  
日本電信電話公社入社。自然言語処  
理の研究に従事。



### 文献紹介

#### 92-38 スループット指向環境上の共有バス結合マルチプロセッサ構成要素の評価

Men-Chow Chiang and Gurindar S. Sohi : Evaluating Design Choices for Shared Bus Multiprocessors in a Throughput-Oriented Environment

[*IEEE Trans. Comput.* Vol. 41, No. 3, pp. 297-317 (Mar. 1992)]

Key: Cache block size, cache set associativity, circuit switched buses, mean value analysis, shared bus multiprocessors, split transaction pipelined buses, trace-driven simulation.

小・中規模マルチプロセッサシステムを実現する場合、複数のプロセッサとメモリをバスで結合する共有バス結合方式は、簡単でかつ安価な実現方法であり、この方式で多くの実験・商用マルチプロセッサシステムが実現されている。本論文では、スループット指向環境における、上記のマルチプロセッサシステムのキャッシュ構成方法やバススイッチ方法等をモデル解析によって評価している。ここでいうスループット指向環境とは、各タスクが1つのプロセッサで実行されている、多重プログラミング環境を示し、マルチプロセッサの性能は、そのマルチプロセッサ全体のスループ

ットで評価される。

評価対象のマルチプロセッサの構成を、図-1に示す。各プロセッサは、キャッシュメモリをもっており、そのキャッシュメモリが共有バスで共有メモリと結合されている。このようなメモリシステムをもつマルチプロセッサの全体的な性能は、メモリシステムの構成に大きく影響され、プロセッサからみた平均メモリアクセス時間  $T_m^p$  は、

$$T_m^p = T_c^p + M \times T_m^c \quad (1)$$

として知られている。ここで、 $T_c^p$ 、 $M$ 、 $T_m^c$  は、それぞれキャッシュのアクセス時間、キャッシュのミスヒット率、キャッシュミスを補うためにかかる平均の時間である。 $T_c^p$  と  $M$  は、キャッシュの構成によって決まる値であり、ユニプロセッサとマルチプロセッサとで、同じ値となる。しかし  $T_m^c$  は、ユニプロセッサとマルチプロセッサでは異なる値となる。なぜならマルチプロセッサの  $T_m^c$  は、バスアクセスのためのキューイングディレイを含んでいるからである。

このようなメモリシステムを持つマルチプロセッサを解析するために、本論文では文献<sup>1)</sup>で提案されている Customized Mean Value Analysis (CMVA) 法を用いてモデル解析を行っている。CMVA 法は、1) 解析モデルを決定する、2) そのモデルの入出力パラメータを決定する、3) (1) 式に代表される平均時間式を入出力パラメータを使って作り、それを応答時間式 (Response Time Equations) と待ち時間式 (Waiting Time Equations) とし、4) 3) の各式から応答時間と待ち時間を見積もる、というものである。本論文では、CMVA 法で見積もった応答時間や待ち時間からスループ