TEXTS AND MONOGRAPHS IN COMPUTER SCIENCE

MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

Volume I: Sets, Relations, and Induction

Peter A. Fejer
Dan A. Simovici

Springer-Verlag

Mathematical Foundations of Computer Science

Volume I:

Sets, Relations, and Induction

Peter A. Fejer
Dan A. Simovici

With 36 Illustrations

Springer-Verlag
New York Berlin Heidelberg London
Paris Tokyo Hong Kong Barcelona

Peter A. Fejer
Department of Math
and Computer Science
University of Massachussetts
at Boston
Boston, MA 02125
USA

Dan A. Simovici
Department of Math
and Computer Science
University of Massachusetts
at Boston
Boston, MA 02125
USA

Series Editor
David Gries
Department of Computer Science
Cornell University
Ithaca, NY 14853
USA

Library of Congress Cataloging-in Publication Data
Fejer, Peter A.
Mathematical foundations of computer science / Peter A. Fejer, Dan A. Simovici.
p. cm. -- (Texts and monographs in computer science)
Includes bibliographical references and index.
Contents: v. 1. Sets, relations, and induction.
ISBN 0-387-97450-4. -- ISBN 3-540-97450-4 (Berlin)
1. Computer science -- Mathematics. I. Simovici, Dan A.
II. Title. III. Series
QA76.9.M35F45 1990
004'.01'51--dc20 90-49595

Printed on acid-free paper.

Photocomposed copy prepared from the authors' LaTeX file.
Printed and bound by R.R. Donnelley and Sons, Harrisonburg, Virginia.
Printed in the United States of America.

9 8 7 6 5 4 3 2 1

ISBN 0-387-97450-4 Springer-Verlag New York Berlin Heidelberg
ISBN 3-540-97450-4 Springer-Verlag Berlin Heidelberg New York

Texts and Monographs in Computer Science

Texts and Monographs in Computer Science

Suad Alagic
Object-Oriented Database Programming
1989. XV, 320 pages, 84 illus.

Suad Alagic
Relational Database Technology
1986. XI, 259 pages, 114 illus.

Suad Alagic and Michael A. Arbib
The Design of Well-Structured and Correct Programs
1978. X, 292 pages, 68 illus.

S. Thomas Alexander
Adaptive Signal Processing: Theory and Applications
1986. IX, 179 pages, 42 illus.

Michael A. Arbib, A.J. Kfoury, and Robert N. Moll
A Basis for Theoretical Computer Science
1981. VIII, 220 pages, 49 illus.

Friedrich L. Bauer and Hans Wössner
Algorithmic Language and Program Development
1982. XVI, 497 pages, 109 illus.

Kaare Christian
A Guide to Modula-2
1986. XIX, 436 pages, 46 illus.

Edsger W. Dijkstra
Selected Writings on Computing: A Personal Perspective
1982. XVII, 362 pages, 13 illus.

Edsger W. Dijkstra and Carel S. Scholten
Predicate Calculus and Program Semantics
1990. XII, 220 pages

W.H.J. Feijen, A.J.M. van Gasteren, D. Gries, and J. Misra, Eds.
Beauty Is Our Business: A Birthday Salute to Edsger W. Dijkstra
1990. XX, 453 pages, 21 illus.

P.A. Fejer and D.A. Simovici
Mathematical Foundations of Computer Science, Volume I: Sets, Relations, and Induction
1990. X, 425 pages, 36 illus.

Melvin Fitting
First-Order Logic and Automated Theorem Proving
1990. XIV, 242 pages, 26 illus.

Nissim Francez
Fairness
1986. XIII, 295 pages, 147 illus.

continued after index

To my parents
P.A.F.

To Doina and Alex
D.A.S.

Preface

As the title suggests, *Mathematical Foundations of Computer Science* deals with those topics from mathematics that have proven to be particularly relevant in computer science. The present volume treats basic topics, mostly of a set-theoretical nature: sets, relations and functions, partially ordered sets, induction, enumerability, and diagonalization. The next volume will discuss topics having a logical nature. Further volumes dealing with algebraic foundations of computer science are also contemplated.

We present the material in a way that is systematic, rigorous, and complete. Our approach is straightforward and, we hope, clear, but we do not avoid more difficult topics or sweep subtle points under the rug. Our goal is to make the subject, as Einstein said, "as simple as possible, but not simpler."

In Chapter 1, we discuss set theory from an intuitive point of view, but we indicate how difficulties arise and how an axiomatic approach might solve these problems.

Chapter 2 presents relations and functions, starting from the notion of the ordered pair. We emphasize the use of relations and functions as structuring devices for data, particularly for relational databases.

In Chapter 3, we provide an introduction to partially ordered sets. We define complete partial orders and prove results about fixed points of continuous functions, which are important for the semantics of programming languages. In the final section of the chapter, we analyze Zorn's Lemma. This proposition may appear to be of remote interest for computer science; nevertheless, results of real interest in computer science, such as connections between various types of partially ordered sets and fixed point results, are based on the use of this lemma.

Chapter 4 is dedicated to the study of mathematical induction. We present several versions of induction: induction on the natural numbers, inductively defined sets, well-founded induction, and fixed-point induction. Mathematical objects, such as formulas of propositional logic, grammars, and recursive functions, important for computer science, receive special attention in view of the role played by induction in their study.

In Chapter 5, we examine mathematical tools for investigating the limits of the notion of computability. We concentrate on diagonalization, a proof method originating in set theory that is an essential tool for obtaining limitative results in the theory of computation.

This volume is organized by mathematical area, which means that material on the same Computer Science topic appears in more than one place. Readers will find useful applications in algorithms, databases, semantics of programming languages, formal languages, theory of computation, and program verification.

There are few specific mathematical prerequisites for understanding the material in this volume, but it is written at a level that assumes the mathematical maturity gained from a good mathematics or computer science undergraduate major. Many of the applications require some exposure to introductory computer science.

Each chapter contains a large number of exercises, many with solutions (which we regard as supplements).

We would like to thank Lynn Montz, Suzanne Anthony and Natalie Johnson of Springer-Verlag for their attention to our manuscript and Karl Berry, Rick Martin, and James Campbell of the Computer Science Laboratory at UMass-Boston for maintaining the systems which allowed us to produce this book. Finally, the authors would like to acknowledge the many judicios remarks and suggestions made by Professor David Gries.

Contents

1

Elementary Set Theory

1.1 Introduction

The concept of set and the abstract study of sets (known as set theory) are cornerstones of contemporary mathematics and, therefore, are essential components of the mathematical foundations of computer science. For the computer scientist, set theory is not an exotic, remote area of mathematics but an essential ingredient in a variety of disciplines ranging from databases and programming languages to artificial intelligence.

Set theory as it is used by working mathematicians and computer scientists was formulated by Georg Cantor[1] in the last quarter of the 19th century. Cantor's approach led to difficulties that we will mention briefly in this chapter. The apparent solution to these difficulties requires an axiomatic approach, which we will allude to but not cover in detail.

In this chapter, we discuss sets and membership and examine ways of defining sets. Then, we introduce methods of building new sets starting from old ones and study properties of these methods.

1.2 Sets, Members, Subsets

Cantor attempted to define the notion of *set* as a collection into a whole[2] of definite, distinct objects of our intuition or our thought. The objects are called *elements* or *members* of the set.

---

[1]The German mathematician Georg F. L. P. Cantor was born on March 3, 1840, in St. Petersburg, Russia, and died on January 6, 1918, in Halle, Germany. He was affiliated with the University of Halle beginning in 1869. Cantor's main contribution was the initial development of modern mathematical set theory.

[2]Zusammenfassung zu einem Ganzen.

This definition does not satisfy the normal requirements of logic, which insist that a newly defined concept be a particularization of an already defined more general concept. Indeed, the term "collection" used in the Cantorian definition is hardly different from the defined term "set." Therefore, we shall regard the notion of set as being a primitive concept, i.e., a general notion that is understood intuitively but not defined precisely and can be used in defining other more particular notions. Hence, motivated by Cantor's "definition," we adopt the rather vague idea that a set is a collection of "things" that are called the elements of the set.

The primary concepts on which set theory is based are set and membership. Since we are viewing a set as a collection of objects, for any set S and object a, either a is one of the objects in S or it is not. In the former case, we use any of the following phrases: "a is a member of S," "a is an element of S," "a is contained in S," or "S contains a," and we write $a \in S$. This use of the symbol $\in$ was introduced by the Italian mathematician Giuseppe Peano[3] because the symbol $\in$ is similar to the first letter of the Greek word ἐστὶ which means "is." We write $a \notin S$ to denote that a is not a member of the set S.

Note that we did not consider the notion of "object" among the primary notions of set theory. Mathematicians have found that mathematics can be developed based on set theory, assuming that every element of every set is itself a set. For example, we shall see later (in Chapter 4) that every natural number can be considered to be a set. A similar point of view can be adopted for all the other common mathematical objects. Therefore, no other objects than sets need be considered, and when we use the term "object," this term can be interpreted to mean "set."

Sometimes, in order to emphasize that the elements of a set $\mathcal{C}$ are themselves sets, we refer to $\mathcal{C}$ as a *collection* of sets.

Two sets are the same if they have the same elements. Although this fact seems intuitively clear, it is important enough to be singled out as a principle of set theory.

> **Principle of Extensionality.** Let S and T be two sets. If for every object a we have $a \in S$ if and only if $a \in T$, then $S = T$.

Sets can be specified in several ways. One method is to list explicitly the members of the set. If $x_1, \ldots, x_n$ are the elements of S, we denote S by

$$\{x_1, \ldots, x_n\}.$$

[3]Giuseppe Peano was born on August 27, 1858, in Cuneo, Italy, and died on April 20, 1932, in Turin. He taught mathematics at the University of Turin starting in 1884. Peano is one of the founders of symbolic logic and made important contributions to the general theory of functions. His main work, *Formulario Mathematico*, published between 1894 and 1908, was an inspiration for further work in the foundations of mathematics done by Russell and Whitehead and the Bourbaki group.

This notation is justified by the principle of extensionality because any other set that has the same elements is the same set. For instance, consider the set whose members are $1, 4, 9$, and 16. We denote this set by

$$\{1, 4, 9, 16\}.$$

If we cannot explicitly list all of the elements of a set we can use various suggestive extensions of the notation just given. For example, the set $\mathbf{N}$ of natural numbers can be denoted by

$$\{0, 1, 2, \ldots\}.$$

On the other hand, a set can also be specified by stating a *characteristic condition*, that is, a condition satisfied by all members of S and not satisfied by any other object. Consider, for instance, the condition

> n is a natural number less than 20 that is a perfect square.

It is easy to see that this is a characteristic condition for the members of the set S defined above: all members of S satisfy the condition, and every object that satisfies the condition is a member of S. Also, note that we may have several characteristic conditions for a set. For instance, the set S can alternatively be specified as consisting of those sums less than 20 of consecutive odd natural numbers starting with 1.

By the principle of extensionality

1. the set $\{1, 4, 9, 16\}$,
2. the set that consists of all natural numbers that are perfect squares and are less than 20, and
3. the set of natural numbers that are sums of consecutive odd natural numbers starting with 1 and are less than 20,

are the same.

In (2) and (3) we have already used implicitly another principle, namely, the

> **Principle of Abstraction.** Given a condition that objects satisfy or do not satisfy, there is a set that consists of the objects that satisfy the condition.

If $\mathcal{K}$ is a characteristic condition that allows us to define the set S, we could denote S by

$$\{x \text{ such that } x \text{ satisfies } \mathcal{K}\}.$$

In practice, we replace the phrase "such that x satisfies" by "|" and thus we write

$$\{x \mid \mathcal{K}\}.$$

The principle of abstraction is a working tool for everyday mathematics. However, its unrestrained application generates contradictions. Suppose, for instance, that, using this principle, we attempt to define the "set" R of all sets that are not elements of themselves,

$$R = \{x | x \notin x\}.$$

We can ask whether R belongs to itself. There are two possible answers:

1. $R \in R$ or
2. $R \notin R$,

and we can prove that both yield contradictions. Indeed, in the first case, the definition of the set R implies that $R \notin R$, which conflicts with the premise of this case. In the second case, the same definition implies $R \in R$; hence, we again obtain a contradiction.

The fact that the principle of abstraction allows this definition, which leads to an immediate contradiction, is known as Russell's paradox after the logician and philosopher Bertrand Russell [4] who discovered it.

Logicians have formulated a more restrictive version of this principle, which appears to eliminate these difficulties, namely, the

> **Principle of Comprehension.** Given a condition that objects satisfy or do not satisfy, and a set U, there is a set S that consists of the elements of U that satisfy the condition.

The difference between the principles of abstraction and comprehension is that in the latter we build a new set starting from an existing set U by collecting those members of U that satisfy the characteristic condition, while in the former we collect all objects satisfying a characteristic condition without restricting the search to the members of a set.

When using the principle of comprehension, we denote the set of those elements of U that satisfy the condition $\mathcal{K}$ by

$$\{x \in U \mid \mathcal{K}\}.$$

[4] Bertrand A. W. Russell, 3rd Earl Russell, was born in 1872 in Trelleck, Monmouthshire ,and died in 1970 near Penrhyndendraeth, Marioneth, in Wales. Russell was one of the major figures of 20th-century philosophy. His work is especially important for philosophical logic and for the theory of knowledge. His most important mathematical work, *Principia Mathematica*, written with A. N. Whitehead, was published between 1910 and 1913.

For instance, we could denote the set S considered above either as

$$\{n \in \mathbf{N} \mid n \text{ is a perfect square and } n < 20\}$$

or

$$\{n \in \mathbf{N} \mid n \text{ is a sum of consecutive odd natural numbers starting with 1 and } n < 20\}.$$

Note that using the principle of comprehension, one cannot duplicate Russell's paradox because of the need to circumscribe the definition of R to the elements of some set U. In fact, for each set U, one can define the set

$$R_U = \{x \in U \mid x \notin x\};$$

however, the existence of this set does not lead to immediate contradiction. If $R_U \in R_U$, then $R_U \in U$ and $R_U \notin R_U$, which is impossible, but if $R_U \notin R_U$, then *either* $R_U \notin U$ *or* $R_U \in R_U$, and this is not contradictory; we merely conclude that $R_U \notin U$ (and $R_U \notin R_U$). Note that we have just shown that for any set U there is another set (namely, R_U) that is not a member of U. Consequently, there is no universal set V such that every set is a member of V.

In formal set theory, the principle of abstraction is rejected, because it leads to Russell's paradox, and the principle of comprehension is used instead. The paradoxical nature of the "set" R is taken to show that there is no such set. Of course, using the principle of comprehension requires, in some cases, additional principles to assert the existence of the set U. This leads to very tedious arguments. To avoid such arguments for now, we continue to use the principle of abstraction; however, in all of our arguments, the use of the principle of abstraction can be replaced by the principle of comprehension plus additional set existence principles.

Definition 1.2.1 *If S and T are two sets such that every element of S is an element of T, then we say that S is* included *in T or that S is a* subset *of T, and we write $S \subseteq T$.*

If S is not a subset of T, we write $S \not\subseteq T$.

Theorem 1.2.2 *For any sets S, T, and U,*

1. *$S \subseteq S$;*
2. *if $S \subseteq T$ and $T \subseteq S$, then $S = T$;*
3. *if $S \subseteq T$ and $T \subseteq U$, then $S \subseteq U$.*

Proof: Suppose that $S \subseteq T$ and $T \subseteq S$. Then, every element of S is an element of T, and every element of T is an element of S. This means that S and T have the same elements, and hence, by the principle of extensionality,

$S = T$. This shows the second part of the theorem. The other two parts are even easier and are left to the reader. ∎

Part (2) of Theorem 1.2.2 provides the standard way of showing that two sets are equal; namely, show that each set is a subset of the other.

Definition 1.2.3 *If S and T are sets such that $S \subseteq T$ and $S \neq T$, then we say that S is* strictly included *in T. We shall denote this by $S \subset T$. If S is not strictly included in T, we write $S \not\subset T$.*

Theorem 1.2.4 *For any sets S, T, and U,*

1. *$S \not\subset S$;*
2. *if $S \subset T$ and $T \subset U$, then $S \subset U$;*
3. *if $S \subset T$, then $T \not\subset S$.*

Proof: Since $S = S$, we cannot have $S \subset S$.

Suppose that $S \subset T$ and $T \subset U$. Then, $S \subseteq T$ and $T \subseteq U$, so, by the third part of the previous theorem, we have $S \subseteq U$. If $S = U$, then we have $S \subseteq T$ and $T \subseteq S$, and hence by the second part of the previous theorem, $S = T$, which contradicts $S \subset T$. Therefore, $S \subset U$.

Finally, suppose that $S \subset T$. If $T \subset S$, then we have $S \subseteq T$ and $T \subseteq S$, which implies $S = T$, contradicting $S \subset T$. ∎

There exists a set with no members. This set is called the *empty set* and is denoted by $\emptyset$. The principle of extensionality implies that there is only one empty set. Furthermore, the empty set is a subset of every set.

If S is a set, we can build a new set by considering the set whose unique member is S. We denote this set by $\{S\}$, and we refer to it as a *singleton*. In particular, we can form the set $\{\emptyset\}$, and this set is not empty since it contains $\emptyset$ as a member.

Definition 1.2.5 *If S is a set, then the* power set *of S is the set which consists of all the subsets of S. We denote the power set of S by $\mathcal{P}(S)$.*

Example 1.2.6 Let $S = \{a, b, c\}$. Then,

$$\mathcal{P}(S) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\}\}.$$

We also have

$$\mathcal{P}(\emptyset) = \{\emptyset\}.$$

Note that for every set S, $\emptyset \in \mathcal{P}(S)$, and so $\mathcal{P}(S)$ is never empty.

We wish to define what is meant by a property of the elements of a set. There are two points of view: intensional and extensional. The intensional viewpoint considers a property of the elements of a set to be a characteristic condition that the elements of the set may or may not satisfy. We have seen already that different characteristic conditions may define the same set, i.e., have the same extension. The extensional viewpoint regards a property as being given by its extension, and this is the point of view we adopt.

Definition 1.2.7 *Let S be a set. By a* property *of the elements of S, we mean a subset of S.*

From this perspective, the power set of a set S consists of all properties of the elements of S. Note that we regard *any* subset of a set as giving a property of the elements of the set even if we have no way of expressing a characteristic condition for the property.

If P is a property of the elements of a set S, we frequently use the phrases "$P(x)$ is true" and "$P(x)$ holds" to mean $x \in P$. The phrase "we will show $P(x)$" means "we will show that $x \in P$".

Example 1.2.8 The property of being an odd natural number is given by the set

$$D = \{1, 3, 5, \ldots\}.$$

The property of being an even natural number is given by the set

$$E = \{0, 2, 4, 6, \ldots\}.$$

The property of being either equal to 0 or the sum of two odd natural numbers is also given by E, and therefore from our extensional point of view, these properties are the same, although from the intensional point of view they are different.

Example 1.2.9 Let M be the collection of all people. Having age 40 is a property of the elements of the set M while having average age 40 is a property of the elements of $\mathcal{P}(M)$, that is of the subsets of M. This is an important distinction in databases where one should differentiate between properties of individual objects and properties of aggregates of objects.

In addition to the notation **N** introduced for the set of natural numbers, we introduce here notations for several other important sets that we will use throughout this book.

> **P** is the set $\{1, 2, 3, \ldots\}$ of positive natural numbers.
>
> **Z** is the set $\{\ldots, -1, 0, 1, \ldots\}$ of integers.
>
> **Q** is the set of rational numbers.
>
> **R** is the set of real numbers.

Note that

$$\mathbf{P} \subset \mathbf{N} \subset \mathbf{Z} \subset \mathbf{Q} \subset \mathbf{R}.$$