

经 典 原 版 书 库

知识表示

(英文版)

Knowledge Representation

Logical, Philosophical,
and Computational Foundations

JOHN F. SOWA

(美) John F. Sowa 著
印第安纳大学



机械工业出版社
China Machine Press

THOMSON

经典原版书库

知识表示

(英文版)

Knowledge Representation
Logical, Philosophical,
and Computational Foundations

(美) John F. Sowa 著
印第安纳大学



机械工业出版社
China Machine Press

John F. Sowa: Knowledge Representation: Logical, Philosophical, and Computational Foundations (ISBN: 0-534-94965-7).

Original edition copyright © 2000 Brooks/Cole, a division of Thomson Learning, Inc.

First published by Brooks/Cole, an imprint of Thomson Learning, United States of America.

All rights reserved.

Reprinted for the People's Republic of China by Thomson Asia Pte Ltd and China Machine Press under the authorization of Thomson Learning. No part of this book may be reproduced in any form without the express written permission of Thomson Learning Asia and China Machine Press.

本书英文影印版由汤姆森学习出版社与机械工业出版社合作出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书版权登记号：图字：01-2003-2625

图书在版编目 (CIP) 数据

知识表示 (英文版) / (美) 索沃 (Sowa, J. F.) 著. -北京: 机械工业出版社, 2003.5
(经典原版书库)

书名原文: Knowledge Representation: Logical, Philosophical, and Computational Foundation
ISBN 7-111-12149-X

I. 知… II. 索… III. 知识表达 - 高等学校 - 教材 - 英文 IV. TP18

中国版本图书馆CIP数据核字 (2003) 第0366051号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 杨海玲

北京瑞德印刷有限公司印刷 · 新华书店北京发行所发行

2003年5月第1版第1次印刷

787mm × 1092mm 1/16 · 38.5印张

印数: 0 001-2000册

定价: 69.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域中取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及收藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国

家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

电子邮件：hzedu@hzbook.com

联系电话：(010) 68995264

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元
石教英
张立昂
邵维忠
周克定
郑国梁
高传善
裘宗燕

王 珊
吕 建
李伟琴
陆丽娜
周傲英
施伯乐
梅 宏
戴 葵

冯博琴
孙玉芳
李师贤
陆鑫达
孟小峰
钟玉琢
程 旭

史忠植
吴世忠
李建中
陈向群
岳丽华
唐世渭
程时端

史美林
吴时霖
杨冬青
周伯生
范 明
袁崇义
谢希仁

Dedication

*To the spirits of the great knowledge engineers,
Aristotle, Leibniz, Kant, Peirce, and Whitehead.*

Preface



Socrates said he was the midwife to his listeners, i.e., he made them reflect better concerning that which they already knew and become better conscious of it. If we only knew what we know, namely, in the use of certain words and concepts that are so subtle in application, we would be astonished at the treasures contained in our knowledge.

IMMANUEL KANT, “*Vienna Logic*”

Like Socrates, knowledge engineers and systems analysts play the role of midwife in bringing knowledge forth and making it explicit. They display the implicit knowledge about a subject in a form that programmers can encode in algorithms and data structures. In the programs themselves, the link to the original knowledge is only mentioned in comments, which the computer cannot understand. To make the hidden knowledge accessible to the computer, knowledge-based systems and object-oriented systems are built around declarative languages whose form of expression is closer to human languages. Such systems help the programmers and knowledge engineers reflect on “the treasures contained in the knowledge” and express it in a form that both the humans and the computers can understand.

Knowledge representation developed as a branch of *artificial intelligence* — the science of designing computer systems to perform tasks that would normally require human intelligence. But today, advanced systems everywhere are performing tasks that used to require human intelligence: information retrieval, stock-market trading, resource allocation, circuit design, virtual reality, speech recognition, and machine translation. As a result, the AI design techniques have converged with techniques from other fields, especially database and object-oriented systems. This book is a general textbook of knowledge-base analysis and design, intended for anyone whose job is to analyze knowledge about the real world and map it to a computable form.

LOGIC, ONTOLOGY, AND COMPUTATION. Knowledge representation is a multidisciplinary subject that applies theories and techniques from three other fields:

1. *Logic* provides the formal structure and rules of inference.
2. *Ontology* defines the kinds of things that exist in the application domain.

3. *Computation* supports the applications that distinguish knowledge representation from pure philosophy.

Without logic, a knowledge representation is vague, with no criteria for determining whether statements are redundant or contradictory. Without ontology, the terms and symbols are ill-defined, confused, and confusing. And without computable models, the logic and ontology cannot be implemented in computer programs. Knowledge representation is the application of logic and ontology to the task of constructing computable models for some domain.

The readers of this book should have some experience in analyzing a problem, identifying the kinds of things that have to be represented, and mapping them to a computable form. This level of experience can be expected of computer science students. Yet because of the interdisciplinary nature of the subject, the book contains considerable material on philosophy and linguistics. Therefore, it is also suitable for philosophy and linguistics students who have some background in artificial intelligence or computer programming. While writing the book, I have used early drafts in graduate-level courses in computer science at Polytechnic University and in the program on Philosophy and Computers and Cognitive Science at Binghamton University.

EXERCISES. At the end of each chapter, the exercises introduce topics that illustrate, supplement, and extend the main presentation. Instead of emphasizing symbol manipulation, the exercises address the problems of analyzing informal specifications and selecting an appropriate ontology for representing them. In effect, the “word problems,” which usually give high-school algebra students the most difficulty, are closer to the central issues of knowledge representation than the purely technical problems of manipulating symbols. Answers and hints for a representative sample of the exercises are included at the end of the book.

All of the major knowledge representations are discussed, analyzed, and related to logic: rules, frames, semantic networks, object-oriented languages, Prolog, Java, SQL, Petri nets, and the Knowledge Interchange Format (KIF). The two basic notations used for logic are predicate calculus and conceptual graphs. Predicate calculus is the traditional logic notation that students must know in order to read the literature of AI and computer science. Conceptual graphs are a two-dimensional form of logic that is based on the semantic networks of AI and the logical graphs of C. S. Peirce. Both notations are exactly equivalent in their semantics, and instructors may choose to use either or both in lectures and exercises.

Examples in this book are illustrated in several languages, but no prior knowledge of any of them is expected. The emphasis is on the semantic principles underlying all languages rather than the syntactic details of particular languages. Although computer exercises can help to show how the theory is applied, this book can be used without any special computer accompaniment.

ORGANIZATION. Chapter 1 introduces logic through a historical survey, ranging from Aristotle's syllogisms to the modern graphic and algebraic systems. The details of the predicate calculus and conceptual graph notations are summarized in Appendix A. For students who have little or no background in logic, the instructor can spend extra time on Chapter 1 and Appendix A to use this book as an introduction to logic. For more advanced students, the instructor can cover Chapter 1 quickly and spend more time on the topics in later chapters.

Chapter 2, which is the most philosophical in the book, introduces *ontology*, the study of existence. Ontology defines the categories of things that are expressed in the predicates of predicate logic, the slots in frames, the tables of a database, or the classes of an object-oriented system. Logic is pure form, and ontology provides the content that is expressed in that form. Depending on the interests of students and the instructor, this chapter can be surveyed briefly or covered in depth.

Chapter 3 introduces the principles of knowledge representation and their role in adapting logic and ontology to the task of constructing computable models of an application domain. It shows how logic and ontology are embodied in a variety of computational languages. This chapter is central to computer applications, but it can be surveyed for students of linguistics or philosophy.

Chapter 4 presents methods for representing dynamically changing processes and events. Petri nets and dataflow graphs are introduced as supplementary notations, which can be translated either to conventional programming languages or to logic in the predicate calculus or conceptual graph notations. Petri nets serve as a bridge between the procedural programming techniques and the declarative logic-based approach that is emphasized in the other chapters.

Chapter 5 shows how purpose and context affect knowledge representation and the various theories of modal and intentional logic. These theories are applied to the encapsulated objects of O-O systems and to the design of interacting agents. This chapter has the most detailed logical development, but much of it can be skipped for students whose background in logic is weak.

Chapter 6, on "knowledge soup," stresses the limitations of logic. It discusses the vague, uncertain, unanalyzed, and often inconsistent mix of facts, opinions, and rules of thumb that people have in their heads. It presents the techniques for reconciling logic to the unpredictable, continuously variable aspects of reality. These techniques are not rejections of logic, but methods for adapting logic to the complexities of the real world.

Chapter 7 discusses the problems of knowledge sharing and the ongoing efforts related to the ANSI and ISO projects on ontology and conceptual schemes. It illustrates critical issues in using logic-based techniques to facilitate communication and interoperability of heterogeneous computer systems.

The first section of every chapter is more introductory and less technical than the remaining sections, and the first paragraph of every section gives a quick overview of the rest. Therefore, readers can survey any chapter by reading just the

first section and the first paragraph of each remaining section. While skimming through a chapter, readers should glance at the illustrations to get an overview of the topics that are covered.

CAST OF CHARACTERS. Science is a human subject, developed by people who step on each other's toes at least as often as they stand on each other's shoulders. The five philosophers to whom this book is dedicated have been admired and trampled more than most. Their theories and practices are among the best available examples of how logic and ontology can be applied to the representation of knowledge in science, business, and everyday life. For a testimonial to their influence, note the references to them in the index of this book.

As Peirce said, every scientist is deeply indebted to a "community of inquirers" whose contributions, criticisms, and collaboration are essential to the development of the science. While writing this book, I benefited enormously from the overlapping communities in which I participated. Among them are my students and colleagues at SUNY Binghamton and Polytechnic University; the members of the ANSI and ISO working groups on conceptual schemas, ontologies, and the CG and KIF standards, which were chaired by Sandra Perez, Tony Sarris, John Sharp, and Baba Piprani; and the FANTA project at IBM, which included Fan Hsu, Bob Spillers, and Martin van den Berg.

My greatest debt is to the community of the conceptual graph workshops and the International Conferences on Conceptual Structures. Since I don't have the space to list all the participants, I'll just list the organizers of the conferences and the editors of the proceedings: Michel Chein, Walling Cyre, Harry Delugach, Judy Dick, Peter Eklund, Gerard Ellis, John Esch, Jean Fargues, Mary Keeler, Bob Levinson, Dickson Lukose, Guy Mineau, Bernard Moulin, Marie-Laure Mugnier, Tim Nagle, Heather Pfeiffer, Bill Rich, Leroy Searle, Bill Tepfenhart, Eileen Way, and Rudolf Wille. I gratefully thank them and everyone mentioned in the proceedings they edited, which are listed in the bibliography of this book.

My community also includes many people whose contributions are not adequately represented in the above lists: Jaime Carbonell, Norman Foo, Benjamin Grosof, Mike Genesereth, Nicola Guarino, Ed Hovy, Fritz Lehmann, John McCarthy, Michael McCord, Robert Meersman, Julius Moravcsik, Mary Neff, Paula Newman, Paul Rosenbloom, Peter Simons, Doug Skuce, Cora Sowa, and Wlodek Zadrozny. Finally, I thank the editors and staff of Brooks/Cole for their patience in waiting for this book to be finished in December for more Decembers than I would like to admit.

John F. Sowa
Croton-on-Hudson, New York

Contents



Preface XIII

CHAPTER ONE

Logic I

1.1 Historical Background	I
1.2 Representing Knowledge in Logic	11
1.3 Varieties of Logic	18
1.4 Names, Types, and Measures	29
1.5 Unity Amidst Diversity	39

CHAPTER TWO

Ontology 51

2.1 Ontological Categories	51
2.2 Philosophical Background	55
2.3 Top-Level Categories	67
2.4 Describing Physical Entities	78
2.5 Defining Abstractions	89
2.6 Sets, Collections, Types, and Categories	97
2.7 Space and Time	109

CHAPTER THREE

Knowledge Representations 132

3.1 Knowledge Engineering	132
3.2 Representing Structure in Frames	143
3.3 Rules and Data	156
3.4 Object-Oriented Systems	169
3.5 Natural Language Semantics	178
3.6 Levels of Representation	186

CHAPTER FOUR

Processes 206

4.1 Times, Events, and Situations	206
4.2 Classification of Processes	213
4.3 Procedures, Processes, and Histories	217
4.4 Concurrent Processes	223
4.5 Computation	232
4.6 Constraint Satisfaction	239
4.7 Change	245

CHAPTER FIVE

Purposes, Contexts, and Agents 265

5.1 Purpose	265
5.2 Syntax of Contexts	274
5.3 Semantics of Contexts	284
5.4 First-Order Reasoning in Contexts	297
5.5 Modal Reasoning in Contexts	307

5.6 Encapsulating Objects in Contexts	321
5.7 Agents	330

CHAPTER SIX

Knowledge Soup 348

6.1 Vagueness, Uncertainty, Randomness, and Ignorance	348
6.2 Limitations of Logic	356
6.3 Fuzzy Logic	364
6.4 Nonmonotonic Logic	373
6.5 Theories, Models, and the World	383
6.6 Semiotics	394

CHAPTER SEVEN

Knowledge Acquisition and Sharing 408

7.1 Sharing Ontologies	408
7.2 Conceptual Schema	417
7.3 Accommodating Multiple Paradigms	427
7.4 Relating Different Knowledge Representations	438
7.5 Language Patterns	445
7.6 Tools for Knowledge Acquisition	452

APPENDIX A

Summary of Notations 467

A.1 Predicate Calculus	467
A.2 Conceptual Graphs	476
A.3 Knowledge Interchange Format	489

APPENDIX B

Sample Ontology 492

B.1 Principles of Ontology	492
B.2 Top-Level Categories	497
B.3 Role and Relation Types	502
B.4 Thematic Roles	506
B.5 Placement of the Thematic Roles	510

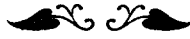
APPENDIX C

Extended Example 513

C.1 Hotel Reservation System	513
C.2 Library Database	515
C.3 ACE Vocabulary	518
C.4 Translating ACE to Logic	518

Answers to Selected Exercises 522**Bibliography 549****Name Index 574****Subject Index 579****Special Symbols 594**

Logic



The very first lesson that we have a right to demand that logic shall teach us is, how to make our ideas clear; and a most important one it is, depreciated only by minds who stand in need of it. To know what we think, to be masters of our own meaning, will make a solid foundation for great and weighty thought.

CHARLES SANDERS PEIRCE, "How to Make Our Ideas Clear"

1.1 Historical Background

The words *knowledge* and *representation* have provoked philosophical controversies for over two and a half millennia. In the fifth century B.C., Socrates stirred up some of the deepest controversies by claiming to know very little, if anything. By his relentless questioning, he destroyed the smug self-satisfaction of people who claimed to have knowledge of fundamental subjects like Truth, Beauty, Virtue, and Justice. By recreating Socrates' dialectical process of questioning, his student Plato established the subject of *epistemology* — the study of the nature of knowledge and its justification. Epistemology, in those days, was literally a matter of life and death. For his alleged impiety in questioning cherished beliefs, Socrates was condemned to death as a corrupter of the morals of Athenian youth.

TERMINOLOGY. Plato's student Aristotle shifted the emphasis of philosophy from the nature of knowledge to the less controversial, but more practical problem of representing knowledge. His monumental life's work resulted in an encyclopedic compilation of the knowledge of his day. But before he could compile that knowledge, Aristotle had to invent the words for representing it. He established the initial terminology and defined the scope of logic, physics, metaphysics, biology, psychology, linguistics, politics, ethics, rhetoric, and economics. For all those fields, the terms that he either coined or adopted have become the core of today's international technical vocabulary. Some of them, such as *category*, *metaphor*, and *hypothesis*, are direct borrowings from Aristotle's Greek. Others, such as *quantity*, *quality*, *genus*,

species, *noun*, *verb*, *subject*, and *predicate*, are borrowings of Latin words that were coined for the purpose of translating the Greek. The English word *quality*, for example, comes from Cicero's word *qualitas*. Cicero explained that he coined the word as a translation of the Greek *poiotēs* (what-kind-ness), which "among the Greeks is not a word of the common people, but of the philosophers" (*Academicae Quaestiones* 1, 6, 24). Today Aristotle's words have been so thoroughly absorbed into English that *category* is a common term on TV quiz shows and *quality* is more often used by salesmen than by philosophers.

SYLLOGISMS. Besides his systematic terminology for representing knowledge, Aristotle developed logic as a precise method for reasoning about knowledge. He invented the *sylllogism* as a three-part pattern for representing a logical deduction. Following is an example of a syllogism taken from Aristotle's *Posterior Analytics* (98b5):

If all broad-leaved plants are deciduous,
and all vines are broad-leaved plants,
then all vines are deciduous.

The basic pattern of a syllogism combines two premises to derive a conclusion. In this example, "all broad-leaved plants are deciduous" is called the *major premise*; "all vines are broad-leaved plants" is called the *minor premise*; and "all vines are deciduous" is the conclusion. Although this example uses words in a natural language, Aristotle presented most of his syllogisms in a highly formalized style, as in the following quotation from the *Prior Analytics* (25b38):

For if A is predicated of every B and B of every C, it is necessary for A to be predicated of every C (for it was stated earlier what we mean by the words of every [*kata pantos*]). Similarly, if A is predicated of no B, and B of every C, it is necessary that A will apply to no C.

With his patterns for syllogisms, Aristotle introduced the first use of variables in history. But he did much more than give a few examples. He presented many pages of systematic analyses with formal *rules of inference* — rules for converting one pattern into another while preserving truth. In the above quotation, Aristotle used terms like *kata pantos* in a technical sense, which he had to explain even for native speakers. Modern symbolic logic uses symbols like \forall instead of words like *kata pantos*, but many programming languages and rule-based expert systems still follow Aristotle's practice of using stylized natural language with variables.

SCHOLASTIC LOGIC. The medieval Scholastics named and classified Aristotle's syllogisms to make them easier to remember. They assigned the vowels A, I, E, and O to the four basic types of propositions: