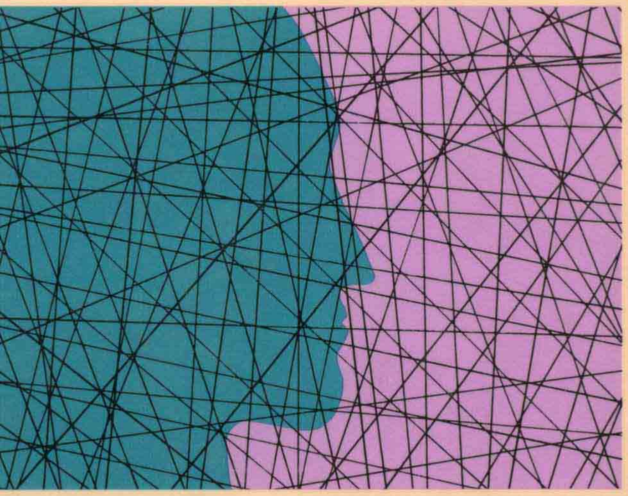
Principles of Semantic Networks

EXPLORATIONS IN THE REPRESENTATION OF KNOWLEDGE



Edited by John F. Sowa

PRINCIPLES OF SEMANTIC NETWORKS

EXPLORATIONS IN THE REPRESENTATION OF KNOWLEDGE

Contributors:

Alexander Borgida
Ronald J. Brachman
Michael J. Coombs
J. M. Crawford
Roger T. Hartley
Barbara Hayes-Roth
Rattikorn Hewett
Paul S. Jacobs
Alfred Kobsa
Benjamin Kuipers
Hector J. Levesque
Robert Mac Gregor
Deborah L. McGuinness
Bernhard Nebel
Lori Alperin Resnick
Peter F. Patel-Schneider
Lenhart K. Schubert
Bart Selman
Stuart C. Shapiro
Lokendra Shastri
Douglas Skuce
Lynn Andrea Stein
Richmond H. Thomason
David S. Touretzky
Robert Wilensky
W. A. Woods
Wlodek Zadrozny

EDITED BY **JOHN F. SOWA**, IBM Systems Research

MORGAN KAUFMANN PUBLISHERS, INC.
SAN MATEO, CALIFORNIA

THE MORGAN KAUFMANN SERIES IN REPRESENTATION AND REASONING

Series editor, Ronald J. Brachman (AT&T Bell Laboratories)

BOOKS

James Allen, James Hendler, and Austin Tate, editors
Readings in Planning (1990)

James F. Allen, Henry Kautz, Richard Pelavin, and Josh Tenenberg
Reasoning About Plans (1991)

Ronald J. Brachman and Hector Levesque, editors
Readings in Knowledge Representation (1985)

Ernest Davis
Representations of Commonsense Knowledge (1990)

Matthew L. Ginsberg, editor
Readings in Nonmonotonic Reasoning (1987)

Judea Pearl
Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (1988)

Glenn Shafer and Judea Pearl, editors
Readings in Uncertain Reasoning (1990)

John Sowa, editor
Principles of Semantic Networks: Explorations in the Representation of Knowledge (1991)

Daniel S. Weld and John de Kleer, editors
Readings in Qualitative Reasoning about Physical Systems (1990)

David E. Wilkins
Practical Planning: Extending the Classical AI Planning Paradigm (1988)

PROCEEDINGS

Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference (KR 91)
edited by Ronald J. Brachman, Hector J. Levesque, and Raymond Reiter (1989)

The Frame Problem in Artificial Intelligence: Proceedings of the 1987 Conference
edited by Frank M. Brown (1987)

Reasoning about Actions and Plans: Proceedings of the 1986 Workshop
edited by Michael P. Georgeff and Amy L. Lansky (1987)

Proceedings of the Second Conference (TARK 1988)
edited by Moshe Y. Vardi (1988)

Proceedings of the Third Conference (TARK 1990)
edited by Rohit Parikh (1990)

Sponsoring Editor *Michael B. Morgan*
Production Editor *Sharon E. Montooth*
Cover Designer *Gary Head*
Electronic Copyediting and Composition *Ocean View Technical Publications*
Text Programming *Bruce Boston*
Proofreader *Kathleen McClung*
Index *Linda Fetters*

Library of Congress data is available.
ISBN 1-55860-088-4
MORGAN KAUFMANN PUBLISHERS, INC.
Editorial Office:
2929 Campus Drive
San Mateo, California 94403
(415) 578-9911

PRINCIPLES OF SEMANTIC NETWORKS

EXPLORATIONS IN THE REPRESENTATION OF KNOWLEDGE

PREFACE

Graphic notations for knowledge have been used for centuries in logic, philosophy, psychology, and linguistics. In the 1960s, the early days of artificial intelligence, network notations were among the first knowledge representation schemes to be developed. They were especially popular for natural language processing because they clarified linguistic relationships that other notations tended to obscure. By the late 1970s, the family of semantic networks had reached a high degree of sophistication and logical rigor.

During the 1980s, the proliferation of commercial applications of AI caused attention to shift to linear notations: most applications used languages like Lisp and Prolog or rule-based expert systems, which have notations that can easily be typed on a keyboard. By the end of that decade, however, two developments rekindled interest in networks: the ubiquity of graphical user interfaces made it possible to show networks on the screen; and the increasing size and complexity of applications made it desirable to have graphic ways of organizing and displaying the contents of a knowledge base. Today, many expert system shells supplement their rule-based and frame-based notations with a semantic network that shows the hierarchy of object types and subtypes.

Developments in related fields of computer science have also enhanced the interest in semantic networks. Procedural object-oriented languages, such as Simula, Smalltalk, and C++, have type or class hierarchies that parallel the hierarchies in semantic networks. Database designers have been using graphic systems for drawing entity-relationship diagrams, which are simplified versions of the kinds of networks used in AI. Many of the issues that those developers are encountering are ones that have long been addressed in the AI research on semantic networks. In particular, recent attempts to apply the object-oriented style of programming to database systems have run into difficulties in reconciling the procedural languages with the purely declarative databases. Semantic networks can help to bridge that gap: like the object-oriented languages, they have type hierarchies with inheritance; and like the database systems, they are purely declarative. In fact, some of the authors who contributed to this book prefer to characterize semantic networks as *object-centered knowledge representations*.

Despite the importance of the subject, there was no book that adequately covered the current theory and applications of semantic networks. To remedy that lack, this book was written as a collaboration with some of the leading researchers in the field. Most of the authors are professors of computer science with long experience in teaching the subject to advanced undergraduate and beginning graduate students. In writing their chapters, they organized the material as they would like to present it to their students. Although each chapter leads the reader to the forefront of research in its area, it starts with systematic definitions and presents the material in an accessible format. The general prerequisites for reading the book are a knowledge of logic and an introductory course on artificial intelligence. The book could be used in a course on knowledge representation or a seminar on semantic networks. Most of the chapters are suitable as supplementary reading for related areas of cognitive science, including linguistics, philosophy, and cognitive psychology.

The idea for the book grew out of a conversation between Norm Sondheimer and John Sowa. We realized that an impressive body of research and development had accumulated on various aspects of semantic networks, but that it was scattered throughout the AI literature. To bring together the most active researchers in semantic networks, we organized a three-day workshop on Catalina Island in 1989. For a sampling of the lively and stimulating discussion at the workshop, the reader should turn to the opening chapter of this book, which is an edited transcript of the concluding panel discussion. After the workshop, the program committee decided on the organization of this book and the selection of chapters to be written. Each chapter was written during the following year, and each was reviewed by two other authors. Not every participant contributed a chapter, and some chapters have coauthors who were not able to participate in the workshop.

The process of developing this book has helped to make it a cohesive and comprehensive review of the state of the art: the authors held extensive discussions with one another at the workshop; they reviewed each other's chapters; and in many cases, they worked closely with one another and the editor to decide what topics to present and how to present them. The chapters fall into three major groups: seven chapters on issues in knowledge representation, which discuss theoretical topics independent of particular implementations; six chapters on formal analyses, which treat the methods of reasoning with semantic networks and their computational complexity; and seven chapters on systems, which show how the theory has been implemented in working systems for knowledge representation.

As the editor, I gratefully acknowledge the collaboration of the workshop organizers and participants, whose help was essential in producing this book and maintaining its quality. Funds for the workshop were provided by a grant from the AAAI and an advance from Morgan Kaufmann Publishers. The general chairman of the workshop was Norm Sondheimer of General Electric Research; the program chairman was John Sowa of the IBM Systems Research Institute; and the local arrangements chairman was Robert MacGregor of the USC Information Sciences

Institute. The program committee members contributed generous amounts of time in helping to plan the workshop, review the preliminary abstracts, and select the talks to be presented and the chapters to be written; they include

- Ron Brachman, AT & T Bell Laboratories,
- Jaime Carbonell, Carnegie Mellon University,
- David Etherington, AT & T Bell Laboratories,
- Norman Foo, Sydney University,
- Christopher Habel, Hamburg University,
- Len Schubert, University of Rochester,
- Stuart Shapiro, State University of New York at Buffalo,
- Robert Simmons, University of Texas,
- Doug Skuce, University of Ottawa,
- James Slagle, University of Minnesota,
- Rich Thomason, University of Pittsburgh,
- Robert Wilensky, University of California at Berkeley.

Finally, this book could not have been produced without the support of Mike Morgan and his able assistants, especially Sharon Montooth, who saw it through to completion.

John F. Sowa
February 1991

CONTENTS

PART I

ISSUES IN KNOWLEDGE REPRESENTATION

A semantic network is a structure for representing knowledge as a pattern of interconnected nodes and arcs. The first semantic networks were implemented in machine translation systems in the early 1960s. Since then, dozens of different versions have been designed and implemented. Although the terminology and notations vary widely, the following themes are common to most of them:

- Nodes in the net represent concepts of entities, attributes, events, and states.
- Arcs in the net, usually called conceptual relations, represent relationships that hold between the concept nodes. Labels on the arcs specify the relation types.
- Some conceptual relations represent linguistic cases, such as *agent*, *patient*, *recipient*, or *instrument*. Others represent spatial, temporal, causal, and logical connectives. Still others specify the role that one entity plays with respect to another, such as *mother*, *owner*, or *residence*; but the representation of roles as relations or concepts is one area of divergence between different systems.
- Concept types are organized in a hierarchy according to levels of generality, such as ENTITY, LIVING-THING, ANIMAL, CARNIVORE, FELINE, CAT. This hierarchy is often called a *type hierarchy* or a *taxonomic hierarchy*. It is also called a *subsumption hierarchy*, since the instances of a general type such as ANIMAL *subsume* the instances of a more specialized type such as CAT.
- Relationships that hold for all concepts of a given type are *inherited* through the hierarchy by all subtypes. Since every animal requires oxygen, the property of requiring oxygen is inherited by every carnivore, feline, and cat.

Despite these common themes, the networks diverge on a number of issues such as philosophical questions of meaning, methods for representing all the quantifiers and operators of logic, techniques for manipulating the networks and performing inferences, and stylistic conventions for drawing the nodes and arcs and labeling them with words or other symbols. Some systems emphasize the ability to assert propositions and reason with them, and others place more emphasis on ways of defining new concepts in the type hierarchy. Some are designed for representing natural language, and others are designed for expert systems applications. Some have a formal basis in logic, while others are much more informal. Despite the differences, their resemblances are sufficient to characterize them as a distinctive family of knowledge representation systems.

SURVEY OF CHAPTERS IN PART I

The chapters in Part I discuss issues in knowledge representation that are independent of any particular implementation. The opening chapter is a transcript of the panel discussion that concluded the Catalina workshop. Although the panel discussion came at the end of the workshop, this chapter belongs at the beginning of the book, since all the other chapters were written after the panel. Its informal style makes it a very readable introduction that displays the motivation for the more formal presentations in later chapters. Among the themes it covers are expressive power vs. computational complexity, formal semantics vs. informal heuristics, methods of integrating knowledge from diverse sources, relationships between theory and applications, relationships of connectionism to semantic networks, and the need for standard test cases or problems that can guide the theoretical studies and measure their success.

The next four chapters address ways in which the knowledge representation supports the reasoning methods. William Woods, the author of Chapter 1, published a classic paper entitled "What's in a Link" in 1975. In it, he criticized the poorly defined semantics of many early networks and he established principles that have helped to guide much of the research in the 1970s and early 1980s. In this book, he analyzes subsumption and taxonomy, two themes from his earlier paper. He discusses their role in knowledge representation and generalizes them to accommodate probabilistic and default rules as well as abstract and partial definitions. Len Schubert also worked with semantic networks in 1975 and was the first to introduce modal operators and definitional mechanisms based on the lambda calculus. In Chapter 2, he argues that the syntax of logic and the inference mechanisms based on it are fundamentally network-like. For that reason, he believes that semantic networks are not competitors to logic, but allied representations that exhibit the underlying logical structure in a perspicuous way. Although Schubert is a strong advocate of network representations, he has reservations about the term "semantic network" since it diverts attention from the fundamental unity between logic and networks. Lokendra Shastri, by contrast, believes that semantic networks are much more than notational variants of other languages or logics. In Chapter 3, he argues that their structure can determine the effectiveness of certain kinds of inferences and the optimal methods for performing computations. In particular, when semantic networks are realized as massively parallel networks, they may provide an appropriate framework for modeling reflexive reasoning—reasoning that can be performed rapidly, effortlessly, and without conscious effort. Stuart Shapiro, who implemented the first semantic networks that could support all of first-order logic, also believes that a properly structured network can support important kinds of "subconscious" reasoning that are not directly representable in the linear form of logic. In Chapter 4, he discusses *cables* and *paths*: a cable represents a set of nodes all linked to a given node by the same relation type; a path is a sequence of arcs through a network. Both of these constructions allow many propositions to be represented implicitly or "subconsciously" and only realized explicitly when there is a specific need for them.

The last two chapters of Part I address aspects of language that are not easily represented in predicate calculus. In Chapter 5, John Sowa argues that a graph logic, such as C. S. Peirce's existential graphs, can represent linguistic structures more faithfully than the predicate calculus. He combines Peirce's graphs with representations from AI and linguistics to form his version of *conceptual graphs*. For a variety of linguistic constructions, he shows that the graphs are simpler than predicate calculus. The differences are most significant in the representation of contexts, indexicals, plurals, and generalized quantifiers. In Chapter 6, Robert Wilensky explores issues related to situation semantics. With a series of examples that illustrate the distinctions between situations and propositions, he shows the inadequacies of various knowledge representations. Some semantic networks, for example, can represent either propositions or situations, but not both. To provide a more general representation for sentences in natural language, Wilensky proposes an extended ontology of situations that clarifies the relationships between different kinds of situations and the propositions that describe them.

GENERAL BACKGROUND

Readers who have had an introductory course in artificial intelligence should be able to read most of the chapters in this book, but they may encounter some unfamiliar terms from philosophy. A word that has become increasingly popular in discussions of knowledge representation is *ontology*, which comes from the Greek *ontos* [being] and *logos* [word]. Ontology is therefore the study of being, or, the basic categories of existence. With the indefinite article, the term *an ontology* is often used as a synonym for a taxonomy that classifies the categories or concept types in a knowledge base. The word *taxonomy* itself comes from the Greek *taxis* [arrangement] and *nomos* [law]. Literally, a taxonomy could be an arrangement based on any kind of law or principle. The most common principle is generalization; in that case, the taxonomy would be a *generalization hierarchy*, more often called a type hierarchy or subsumption hierarchy. A taxonomy could also be based on the part-whole relation. Such an arrangement is called a *meronomy* from the Greek *meros* [part]. Some people use the word *partonomy* for *meronomy*, but the word *part* (from the Latin *pars, partis*) is out of place among all those Greek terms. Another term derived from *meros* is *mereology*, which is the study of parts and wholes and the axioms for relating them. Philosophers and linguists are beginning to consider mereology as an alternative to set theory, since the plurals and mass terms in natural language can be represented more easily in mereology. Chapter 5 includes a brief discussion of mereology. One other term that is often used in AI is *epistemology*, from the Greek *episteme* [knowledge]. In philosophy, epistemology is the study of the limits and validity of knowledge and the criteria that distinguish it from belief. In AI, the term *epistemology* is sometimes applied to the categories of knowledge; but since those categories are the same as the categories of existence, the term *ontology* would be more appropriate.

Logic is also used throughout this book. Len Schubert's observation about the network-like nature of logic has some support from the history of predicate calculus. In 1879, the German philosopher Gottlob Frege used a tree notation for his *Begriffsschrift* [concept writing], which was the first complete system of predicate logic; no one else, however, adopted his notation. In 1883, the American philosopher Charles Sanders Peirce independently developed the linear notation that is used today. Peirce's notation was adopted by the German logician Ernst Schröder. The Italian mathematician Giuseppe Peano adopted the system from Schröder, but changed the symbols. He started the practice of turning letters upside down and backwards to represent logical operators: the letter E for existence became the existential quantifier $\exists$; the letter C for consequence became the implication symbol $\supset$; the letter V for the Latin *vel* [or] became the symbol for disjunction $\vee$; and the V turned upside down became the symbol for conjunction $\wedge$. Meanwhile, Peirce was not satisfied with the linear notation and experimented with networks, which he felt would show the structure of logic more clearly. In 1896, he developed his *existential graphs*, a graphical system for logic with complete rules of inference. In his later work on logic, Peirce mainly used the graphs, which he considered "the logic of the future." The modern interest in semantic networks suggests that Peirce may have been right.

Since most chapters in this book freely use the notation and terminology of propositional and predicate logic, an introduction to logic is a prerequisite. However, many of the terms would not be mentioned in an introductory course. Following are some of them:

- *Monotonic logic* is standard logic. It is called monotonic because the number of provable theorems increases monotonically as the number of assumptions increases. Adding a new axiom can never cause a previous theorem to become unprovable. If the new axiom causes a contradiction, then everything becomes provable.

- *Nonmonotonic logic* is the name for a family of new logics used to represent defaults and exceptions. Tweety the penguin is a commonly used example. If Tweety is a bird, then one might assume that Tweety can fly. But the additional information that Tweety is a penguin should block the proof that Tweety can fly. That kind of blocking, which is characteristic of nonmonotonic logics, is not possible in standard logics. Examples of nonmonotonic logics include *default logic*, logics with *negation as failure*, and logics based on the principle of *circumscription*. Some of the chapters in Part II discuss *path-based* methods for nonmonotonic reasoning in semantic networks.
- *Sorted logic* restricts each variable to a specific sort. In standard logic, a quantifier like $(\forall x)$ is completely unrestricted, and x could range over any entity in the universe. In sorted logic, however, a quantifier like $(\forall x$:DONKEY) limits x to entities of the sort DONKEY. The sorts of sorted logic correspond to the types of a semantic network, and the same kinds of inheritance mechanisms may be used to improve the efficiency of proof procedures.
- *Higher-order logics* allow variables to range over functions and predicates, unlike *first-order logic*, where quantified variables can only range over simple individuals. Allowing quantifiers to range over functions and predicates makes a major increase in expressive power, but at the expense of serious computational overhead. For many purposes, first-order logic is adequate, but sometimes a single statement in higher-order logic can express a generalization that would require infinitely many statements in first-order logic.
- *Fuzzy logic* is a family of logics that have a continuous range of truth values. Instead of the two values *true* and *false*, they allow an arbitrary number of values, such as 1.0 for certainly true, 0 for certainly false, 0.9 for very strong likelihood, 0.7 for mild likelihood, and 0.5 for unknown or indifferent. Fuzzy logic has some affinity with probability theory, and it has important applications in control systems. However, many people have serious reservations about its philosophical foundations.
- *Modal logics* represent modalities, as expressed by the English modal auxiliary verbs, *may, can,* and *must.* The two basic modal operators are *possibility*, often represented by a diamond symbol $\Diamond$, and *necessity*, represented by a small box $\Box$. Other versions of modal logic include *deontic logic*, which represents the two modes *permissible* and *obligatory*.
- *Intensional logics* are closely related to modal logic. They are used to represent *propositional attitudes*, or they are used to represent verbs that express some mental attitude toward a proposition. Such verbs include *know, believe,*

think, hope, wish, fear, and *imagine*. If the only intensional verbs are *know* and *believe*, the logic is called *epistemic logic*.

- *Temporal logics* deal with time, which raises complications that are not handled by the static models of standard logic. Some versions of temporal logic have a close affinity with modal logic, with the □ symbol representing *always* and the ◊ symbol representing *sometimes*. *Tense logics* represent the multiple reference times implied by the tenses in natural languages. A sentence such as *Tom will have been traveling* implies a time t_1 when the sentence is spoken, a later time t_2 when Tom is traveling, and a time t_3 when Tom's travel started.

Every system of logic, whether represented by networks or by linear strings, has a notation that is purely syntactic. Calling something a semantic network does not confer any deeper semantics upon it. To give it semantic content, there must be an independent basis for determining the meaning of its nodes and arcs. In talking about meaning, philosophers have drawn a distinction between the *intension* of a term (its basic meaning in itself) and its *extension* (the set of things it refers to). Frege used the example of the *evening star* vs. the *morning star*. These two terms have different intensions: one means a star that is seen in the morning, and the other means a star that is seen in the evening. Yet both of them have the same extension, namely the planet Venus. A semantic basis could be extensional or intensional. An extensional definition of COW, for example, would be a catalog of all the cows in the world; an intensional definition would specify the properties or criteria for recognizing cows without regard to their possible existence.

The usual semantic basis for logic is an extensional approach called *model theory*, which was originally developed by Alfred Tarski. For propositional logic, model theory reduces to the theory of truth tables, which are covered in most introductory courses. For predicate logic, however, the models must include the entities over which the quantifiers range. To represent those entities, a model is constructed as an abstract data structure with two components: a set of elements called *individuals*, which represent every entity in the domain of discourse, and a set of relations defined over those individuals. Besides the data structure, a model has an *interpretation function* that maps formulas and terms into their *denotations*. The denotation of a formula is a truth value **T** or **F**. The denotation of a constant is an individual: the constant *Mary*, for example, would denote an individual named Mary.

Model theory can be adapted to graphs in a concise and elegant way, since the data structure of a model is naturally graph-like. Any set of individuals and relations can be represented by a graph with the individuals as the nodes and the relations as the arcs; each arc is labeled with the name of the relation. A relation with more than two arguments can be represented by a node for each *n*-tuple of the relation linked by an arc to each individual in that *n*-tuple. When the formulas and the models are both represented by graphs, the interpretation function can be